

**CUT TECHNIQUES IN INTEGER
LINEAR PROGRAMMING**

By

RICHARD MORTON GREENHAW, JR.

Bachelor of Science
Central State University
Edmond, Oklahoma
1970

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1972

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF EDUCATION
July, 1979

Thesis
1979D
G813c
cop. 2

CUT TECHNIQUES IN INTEGER
LINEAR PROGRAMMING



Thesis Approved:

Marvin S. Keener

Thesis Adviser

Jeanne Agnew

William Adair

H.G. Burdard

Norman N. Blushorn

Dean of the Graduate College

1041505

ACKNOWLEDGMENTS

I would like to express my appreciation for the help, insight, and example provided by the late Professor E. K. McLachlan. Without his encouragement this study would not have been started. I would like to express my thanks to Dr. Marvin Keener for his kindness in assuming the responsibility for guiding this project toward its finish. Without his interest and good advice this study could not have been completed.

Additional thanks must also be given to the other committee members, Dr. Jeanne Agnew, Dr. Hermann Burchard, and Dr. Bill Adrian, for their assistance and advice in the preparation of the final manuscript. I would also like to thank Ms. Beverly Gifford for the excellence of the typing in the final copy.

Finally, I wish to express special gratitude to my wife, Becky, and our sons, Joshua and Matthew, for their understanding, encouragement, and many sacrifices.

TABLE OF CONTENTS

Chapter	Page
I. PRELIMINARIES	1
Introduction	1
Definitions and Conventions	3
The Associated Linear Program	8
Approaches to Solving the ILP	12
The Simplex Method	15
II. THE DEVELOPMENT OF THE EARLY CUT TECHNIQUES	34
Relationship to Linear Programming	34
Early Uses of the Cutting-Plane	37
The Cutting-Plane Methods of Gomory	39
III. DEVELOPMENT OF THE LATER CUT TECHNIQUES	63
Primal Methods	63
Deep Cuts	72
Intersection Cuts	79
IV. A CHARACTERIZATION OF CUTS	89
Definitions and Examples	89
Jeroslow's Theorems	91
Characterization of the Cuts	96
A SELECTED BIBLIOGRAPHY	104

LIST OF TABLES

Table	Page
I. Basic Matricies and Basic Solutions for Example 1.1.1 . . .	18
II. The Initial Simplex Tableau	24
III. The Second Simplex Tableau	24

LIST OF FIGURES

Figure	Page
1. Graph of Example 1.1.1	4
2. Graph of Associated LP of Example 1.1.1	10
3. Graph of Example 1.1.1 with Additional Constraints	10
4. Graph of Example 1.1.1 with a Valid Cut	14
5. Basic Solutions of the Associated LP to Example 1.1.1 Plotted with x_1, x_2 - Coordinates	19
6. Tableaux for the Simplex Algorithm Applied to the LP of Example 1.1.1	26
7. Feasible Set and Hyperplanes (Cuts) for Example 2.3.1	45
8. Basic Solutions and Cut-Generating Hyperplanes for Example 2.3.4, First Iteration	58
9. Basic Solutions and Cut-Generating Hyperplanes for Example 2.3.4, Second Iteration	60
10. Solutions and Cuts of Example 3.1.1	71
11. Feasible Region and Cuts of Example 3.2.2	77
12. Intersection Cuts for Example 3.3.1	83
13. The Relationship Between K_1 and K_S	87
14. Illustration for the Proofs of Theorem 3.3.1	88
15. Subadditive Cuts for Example 1.1.1	95
16. The Convex Balls K and C in R^2	101

CHAPTER I

PRELIMINARIES

Introduction

The central problem of linear programming: the optimization of a linear, real-valued function under linear constraints, is commonly and efficiently solved by means of the simplex method. However, the special case in which variables of the linear function must take on only integer values has not admitted a similar, efficient method of solution.

In 1958 R. Gomory (24) developed an algorithm for the solution of the integer programming problem based on the idea of "cutting off" a non-integral, optimal solution by means of an additional constraint. Since that time, the focus of research in the area has centered on the development of techniques to generate such cut constraints, and upon the discovery of algorithms to implement them.

One objective of this paper is to survey the development of techniques for generating cuts. Another is to give descriptions of these techniques motivated by their geometry and illuminated by means of examples. A third objective is to unify these cut techniques by showing how they can be developed from a certain class of functions.

This study will be limited by certain constraints. While several algorithms for solving specific types of integer programming problems have been developed, only methods used to solve general integer programming problems will be discussed. Also, general-purpose methods other

than those using cut techniques will be excluded. Another limitation involves the distinction between techniques used to generate cuts and the algorithms which make use of these techniques. The literature includes surveys of the earlier algorithms (see (20)(5)(6)) while much of the later research effort has involved finding new cut techniques to be used in these algorithms. Therefore, while the nature of these algorithms will be described, the focus of this paper will be upon the techniques for cut generation.

As an illustration and a basis for discussion of the fundamentals of integer linear programming consider the following example:

Example 1.1.1. A small computer training institute is considering the addition of two courses to its curriculum. Course 1 requires eight hours faculty time and one hour machine time per section per week. Course 2 requires three hours faculty time and one hour machine time per section per week.

The institute has available 27 hours of faculty time and five hours of machine time per week. If Course 1 yields a net profit of \$700 per section (at capacity enrollment), and Course 2 yields a net profit of \$300 per section, how many sections of each course should be added in order to maximize the school's profit?

This example can be set up as an integer linear programming problem with two constraints and two variables. Let x_1 represent the number of sections of Course 1 added and x_2 the number of sections of Course 2 added. The problem may then be stated:

$$\begin{aligned} &\text{maximize} && 700x_1 + 300x_2 \\ &\text{subject to} && 8x_1 + 3x_2 \leq 27 \\ &&& x_1 + x_2 \leq 5, \quad x_1 \text{ and } x_2 \geq 0 \text{ and integer.} \end{aligned}$$

Figure 1 is a graph of this problem. The vectors (x_1, x_2) which satisfy the constraints of the problem are the lattice points bounded by the axes and the lines: $8x_1 + 3x_2 = 27$ and $x_1 + x_2 = 5$. The slope of the produced line $700x_1 + 300x_2$ is also shown.

Definitions and Conventions

An integer linear program (ILP) is a mathematical model which is designed to produce one or more vectors which optimize a given function while satisfying a system of constraints. The function to be optimized, the objective function, must be linear, and the optimization may be either maximization or minimization. The system of constraints consists of linear inequality, non-negativity, and integrality constraints.

In general any maximum integer linear program with m constraints and n variables can be stated:

$$\text{maximize} \quad c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1.2.1)$$

$$\text{subject to} \quad a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \quad (1.2.2)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$\text{and} \quad x_1, x_2, \dots, x_n \geq 0,$$

$$x_1, x_2, \dots, x_n \text{ integer.}$$

This maximum ILP can be stated more succinctly in matrix form:

$$\text{maximize} \quad xc$$

$$\text{subject to} \quad Ax \leq b, x \geq 0 \text{ and integer,}$$

where $x = (x_1, x_2, \dots, x_n)$, $c = (c_1, c_2, \dots, c_n)$, $b = (b_1, b_2, \dots, b_m)$,

$A = (a_{ij})$, $x \geq 0$ means $x_i \geq 0$, $i = 1, 2, \dots, n$, and x integer means x_i

integer, $i = 1, 2, \dots, n$.

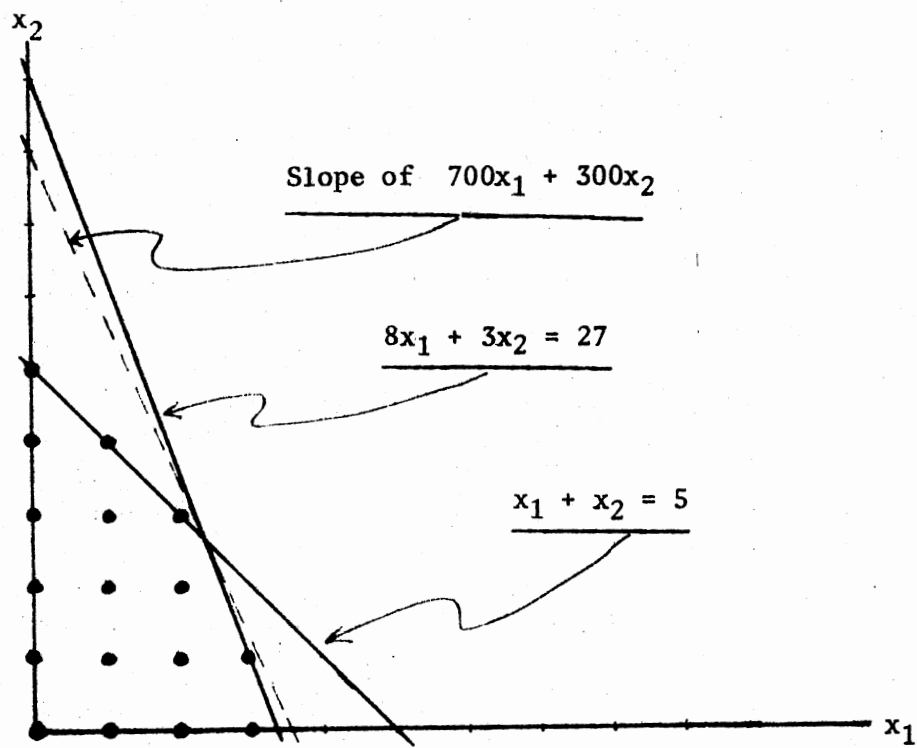


Figure 1. Graph of Example 1.1.1

The ILP of Example 1.1.1 can be written in matrix form:

$$\text{maximize} \quad (x_1, x_2) \cdot (700, 300)$$

$$\text{subject to} \quad \begin{bmatrix} 8 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 27 \\ 5 \end{bmatrix}, \quad (x_1, x_2) \geq 0 \text{ and integer.}$$

It is sufficient to consider only the maximum ILP since any minimum ILP can be written as a maximum problem. For example, consider the following minimum ILP:

$$\text{minimize} \quad xc'$$

$$\text{subject to} \quad A'x \geq b', \quad x \geq 0 \text{ and integer.}$$

By setting $c = -c'$, $b = -b'$, and $A = -A'$ this ILP can be written as:

$$\text{maximize} \quad xc$$

$$\text{subject to} \quad Ax \leq b, \quad x \geq 0 \text{ and integer.}$$

The maximum ILP with inequality constraints (1.2.2) is said to be in standard form. It is sometimes convenient to write these constraints as equations rather than inequalities. To accomplish this a non-negative variable must be added to the left-hand side of each inequality constraint. The ILP resulting from this modification is said to be in canonical form. This canonical form of the maximum ILP can be stated:

$$\text{maximize} \quad c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$\text{subject to} \quad a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2$$

$$\vdots$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m$$

$$\text{and} \quad x_1, x_2, \dots, x_{n+m} \geq 0 \text{ and } x_1, \dots, x_n \text{ integer.}$$

The non-negative variables, $x_{n+1}, x_{n+2}, \dots, x_{n+m}$, added to each inequality of the standard maximum ILP to form the m equations of the

canonical form are referred to as slack variables. In the case of the minimum ILP the left-hand side of each inequality is greater than or equal to its right-hand side. Therefore, in order to transform the standard minimum ILP into a canonical problem negative slack variables should be added to the left-hand side of each constraint. However, if the standard minimum ILP is first transformed into a standard maximum ILP, and this standard maximum problem is then written in canonical form, negative slack variables can be eliminated.

The canonical maximum ILP can be written in matrix form:

$$\text{maximize} \quad x'c' \quad (1.2.3)$$

$$\text{subject to} \quad A'x' = b', \quad x' \geq 0 \text{ and } (x_1, x_2, \dots, x_n) \text{ integer,}$$

where $x' = (x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_{n+m})$, $c' = (c_1, c_2, \dots, c_n, 0, \dots, 0)$, $b' = b$, and $A' = (A : I)$. Example 1.1.1 appears in this form as:

$$\text{maximize} \quad (x_1, x_2, x_3, x_4) \cdot (700, 300, 0, 0)$$

$$\text{subject to} \quad \begin{bmatrix} 8 & 3 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} [x_1, x_2, x_3, x_4]^T = \begin{bmatrix} 27 \\ 5 \end{bmatrix},$$

$$(x_1, x_2, x_3, x_4) \geq 0 \text{ and } (x_1, x_2) \text{ integer.}$$

While the coefficients of c' , b' , and A' in (1.2.3) may take on real values in general, the nature of the problem for which the ILP is a model usually requires only rational coefficients. Example 1.1.1 with its coefficients representing hours of time and dollars of profit is typical. Also, the digital computers on which almost all ILP problems are evaluated admit only rational values for input, output, and computation. Thus, the assumption that the objective function and constraint coefficients be rational is not limiting in any practical sense.

Furthermore, any equation (or inequality) with rational coefficients

can be written as an equation (or inequality) in integral coefficients by multiplying through by a common denominator. This multiplication does not change the nature of the constraints. Their representation is altered, but they remain the same. However, multiplying the objective function by a constant will change its nature. If the objective function $x'c'$ of (1.2.3) is maximized at $\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n+m})$ with value $\bar{x}c'$, then the function $kx'c'$, where k is a positive constant, will also be maximized at \bar{x} , but with a value of $k\bar{x}c'$. Therefore, multiplying the objective function by a positive constant will not change optimal solutions of the maximum ILP, but will cause a change in the value of the objective function. Since this new value will be the constant k times the previous value, the previous value can be determined with little difficulty.

Thus, it can be assumed that any ILP can be written in the form of a canonical maximum ILP with integer coefficients:

$$\text{maximize} \quad xc \quad (1.2.4)$$

$$\text{subject to} \quad (A:I)(x; x_{n+1}, \dots, x_{n+m})^T = b$$

$$\text{and} \quad (x; x_{n+1}, \dots, x_{n+m}) \geq 0 \text{ and } x \text{ integer,}$$

where x and c are n -vectors, b is an m -vector, I is the $m \times m$ identity matrix, A is an $m \times n$ coefficient matrix, and c , b , and A are integral. It may be noted that while the slack variables $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ are not required to be integer valued, they will take on integer values in (1.2.4) when x is integer.

Several applications of linear programming involve questions for which the answers are of the nature "yes" or "no"; for example, an assignment problem asking which worker should be assigned to which job in order to maximize productivity. Such problems are usually formulated as linear programs in these terms: a "yes" answer corresponds to a value $x_i = 1$,

and a "no" to $x_i = 0$. For this reason such problems are referred to as zero-one or binary linear programs. A maximum binary linear program may be stated:

$$\text{maximize} \quad xc \quad (1.2.5)$$

$$\text{subject to} \quad Ax = b \text{ and } x_i = 0 \text{ or } 1, i = 1, 2, \dots, n,$$

where x and c are n -vectors, b is an m -vector, and A is $m \times n$.

It is possible to recast such a maximum binary linear program as a standard maximum ILP by including inequality constraints which force the x_i to be no greater than 1. The binary problem (1.2.5) can then be stated:

$$\text{maximize} \quad xc$$

$$\text{subject to} \quad A'x \leq b', x \geq 0 \text{ and integer,}$$

where $A' = \begin{bmatrix} A \\ \dots \\ I_n \end{bmatrix}$ and $b' = (b; u)$ where u is the n -vector $(1, 1, \dots, 1)$.

The Associated Linear Program

If the integrality condition on the variables x_1 and x_2 of Example 1.1.1 is relaxed the resulting problem is a maximum linear program:

$$\text{maximum} \quad 700x_1 + 300x_2 \quad (1.3.1)$$

$$\text{subject to} \quad 8x_1 + 3x_2 \leq 27 \quad (1.3.2)$$

$$x_1 + x_2 \leq 5$$

$$\text{and} \quad x_1, x_2 \geq 0.$$

The linear program (LP) associated with an ILP is called the associated LP. The associated LP is also referred to as a relaxation of the ILP, because the integrality constraint has been relaxed. The ILP, on the other hand, is often referred to as a restriction of its associated LP. The constraint system of the associated LP (in standard form) consists of only the inequality and non-negativity constraints.

The collection of all vectors which satisfy the constraint system of the associated LP is called the feasible set. A member of the feasible set is called a feasible solution of the associated LP or just feasible. If the feasible set is empty the LP is said to be infeasible. In Example 1.1.1 the feasible set is the set of all vectors bounded by the axes and the lines: $8x_1 + 3x_2 = 27$ and $x_1 + x_2 = 5$. An example of a feasible solution for Example 1.1.1 is the vector $(3/2, 5/2)$.

The feasible set of Example 1.1.1 is closed and bounded and thus the linear objective function will be bounded on the feasible set and assumes its maximum there; that is, there is at least one vector, (\bar{x}_1, \bar{x}_2) , in the feasible set such that $700x_1 + 300x_2 \leq 700\bar{x}_1 + 300\bar{x}_2$ for every vector (x_1, x_2) in the feasible set. In general, the objective function of any LP with a bounded feasible set will be optimal for at least one feasible solution. Any such feasible solution (i.e., one which optimizes the objective function) is called an optimal solution of the LP.

An LP with an unbounded feasible set may not possess an optimal solution. However, the feasible set of an associated LP may be assumed to be bounded. For if the feasible set is unbounded in the direction of the unit vector u_1 , a constraint: $x_1 \leq M$ (M arbitrarily large), may be added to the LP.

Figure 2 shows the feasible set of the associated LP for Example 1.1.1, along with a sequence of parallel lines representing various values of the objective function. Any vector on a given line will give the same value to the objective function. As the objective function increases in value the corresponding line is pushed outward from the origin. The outermost line passes through the feasible solution $\bar{x} = (12/5, 13/5)$. Any line representing a greater value will contain no feasible solution.

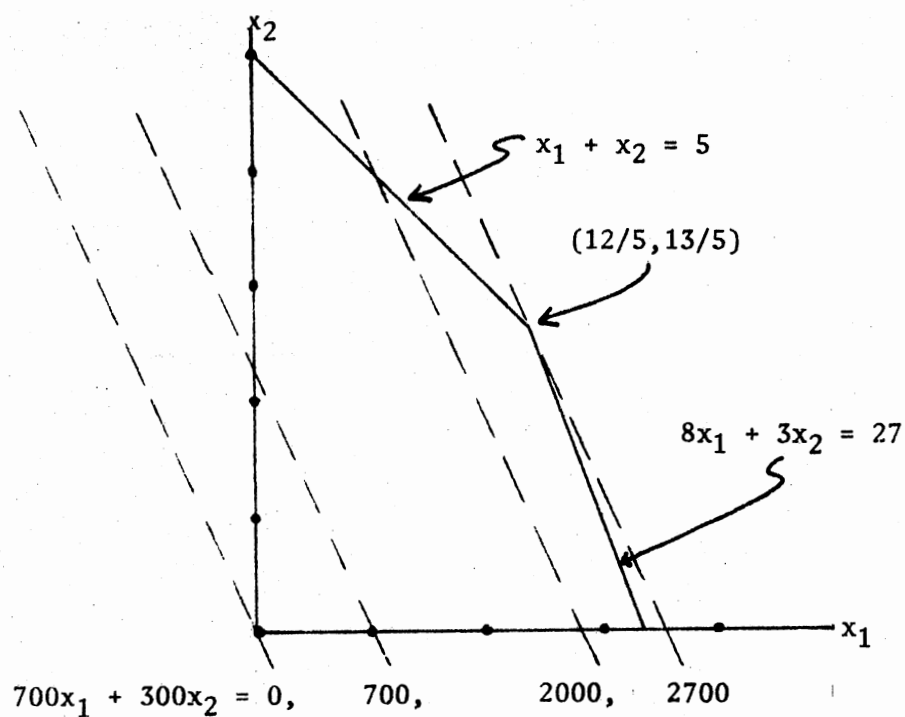


Figure 2. Graph of Associated LP of Example 1.1.1

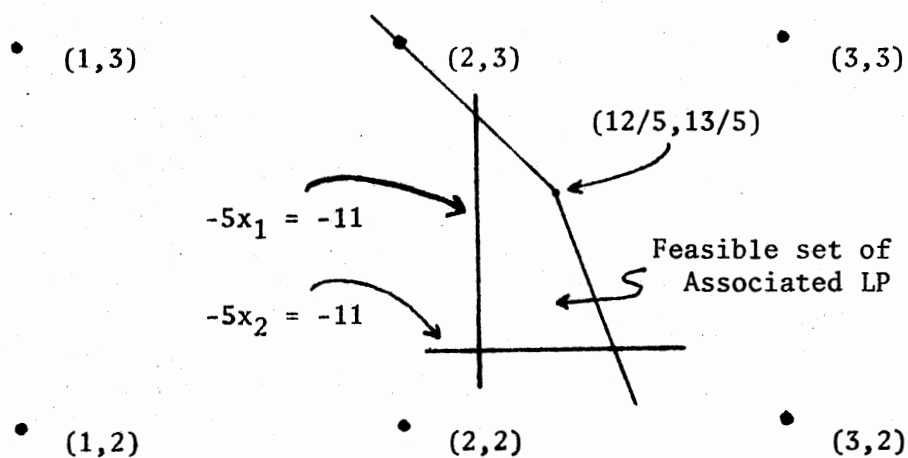


Figure 3. Graph of Example 1.1.1 with Additional Constraints

Thus, $\bar{x} = (12/5, 13/5)$ is an optimal solution of the associated LP. In this case \bar{x} is the only optimal solution. However, had the produced line been parallel to the line $8x_1 + 3x_2 = 27$ any feasible solution on that line would have been an optimal solution.

In general, the bounded feasible set for an LP with m constraints and n variables is a convex polytope in R^n whose faces are contained in the hyperplanes which represent the constraints of the LP. It can be shown (36, p. 22) that any such LP will possess an optimal solution which is an extreme point of this convex polytope. Thus, if only one optimal solution to an associated LP exists (as in Example 1.1.1) it must be an extreme point of the feasible set; and if an optimal solution is found which is not an extreme point, another can be found which is.

A vector which satisfies all the constraints of the ILP (including non-negativity and integrality) is said to be a feasible solution to the ILP or just ILP-feasible. The collection of all such vectors is called the ILP-feasible set. Any ILP-feasible vector which optimizes the objective function of the ILP is said to be an optimal solution of the ILP.

The ILP-feasible set is, therefore, a subset of the feasible set of the associated LP. Thus, the maximum value of the objective function of the ILP will be no greater than that of the associated LP. It may occur that an optimal solution of the associated LP exists while the ILP-feasible set is empty. As an example of this situation consider the ILP of Example 1.1.1 with the additional constraints:

$$-5x_1 \leq -11$$

$$-5x_2 \leq -11.$$

Figure 3 illustrates the feasible set of this associated LP and its optimal solution. In such a case no optimal solution to the ILP exists.

Approaches to Solving the ILP

There are two basic approaches to the problem of isolating an optimal solution from among the vectors of the ILP-feasible set. The first approach is based on the fact that the convex polytope representing the bounded feasible set of the associated LP contains only a finite number of lattice points (i.e., the ILP-feasible vectors). This approach, called enumeration, attempts to list each ILP-feasible vector along with its objective function value. An ILP-feasible vector whose objective function value is optimal is, therefore, an optimal solution to the ILP.

To illustrate this approach consider Example 1.1.1. The 17 ILP-feasible vectors are shown in Figure 1. An examination of these vectors reveals that the vector $(0, 0)$ gives the minimum value to the objective function while the vector $(3, 1)$ gives the maximum value. An optimal solution to this ILP is thus $\bar{x} = (3, 1)$ with an optimal value of 24.

This type of enumeration is primitive in that all ILP-feasible vectors are considered explicitly. Obviously, such a procedure would be of little use with even a medium-sized problem such as one tested by G. T. Martin (38): a 42 city traveling salesman problem (i.e., a problem of finding the shortest route by which a salesman can tour the 42 cities). This problem involved 84 constraints and 861 binary variables; and thus, on the order of 2^{861} ILP-feasible vectors. As a result most enumeration techniques employ methods which implicitly enumerate as many ILP-feasible vectors as possible.

For example, the fact that the objective function of Example 1.1.1 has no negative coefficient would imply that the objective function value of the vector $(2, 2)$ is at least as great as that of $(2, 1)$, $(0, 1)$, $(1, 2)$, $(1, 1)$, $(1, 0)$ or $(0, 0)$; and thus, that these vectors need not be listed explicitly.

The second basic approach to locating an optimal solution to an ILP is based on the concept of generating an additional constraint for the associated LP. First, an optimal solution for the associated LP is found. If this solution is non-integral an additional constraint is generated which eliminates or "cuts off" part of the feasible set including the associated LP's optimal solution. This constraint must not, however, eliminate any ILP-feasible vector from the feasible set. Thus, a new, smaller feasible set results which still contains all ILP-feasible vectors. The objective function is now optimized over this smaller feasible set. If the resulting LP optimal solution is integral, an optimal solution to the ILP has been found; otherwise, the process is repeated.

Techniques which use this approach are called cutting-plane techniques or cut techniques, and any new constraint added to the associated LP which does not eliminate any ILP-feasible vector from the feasible set is termed a valid cutting-plane or simply a valid cut. In general, a cut should be deep enough to eliminate the current LP optimal solution, but not so deep as to cut off any ILP-feasible vector. In addition, the relationship between consecutive valid cuts should be such that the technique will converge to an ILP optimal solution in a finite number of iterations.

Figure 4 shows the feasible set and an optimal solution of the associated LP of Example 1.1.1 along with a valid cut:

$$2x_1 + x_2 \leq 7. \quad (1.4.1)$$

If this cut constraint is appended to the problem, a new optimal solution to the associated LP results. This solution, (3, 1), being integral, must be an optimal solution to the ILP.

The cut (1.4.1) passes through two ILP-feasible vectors. In general this will not be the case; a valid cut need not pass through any ILP-feasible vector. However, cuts like (1.4.1) are desirable because they

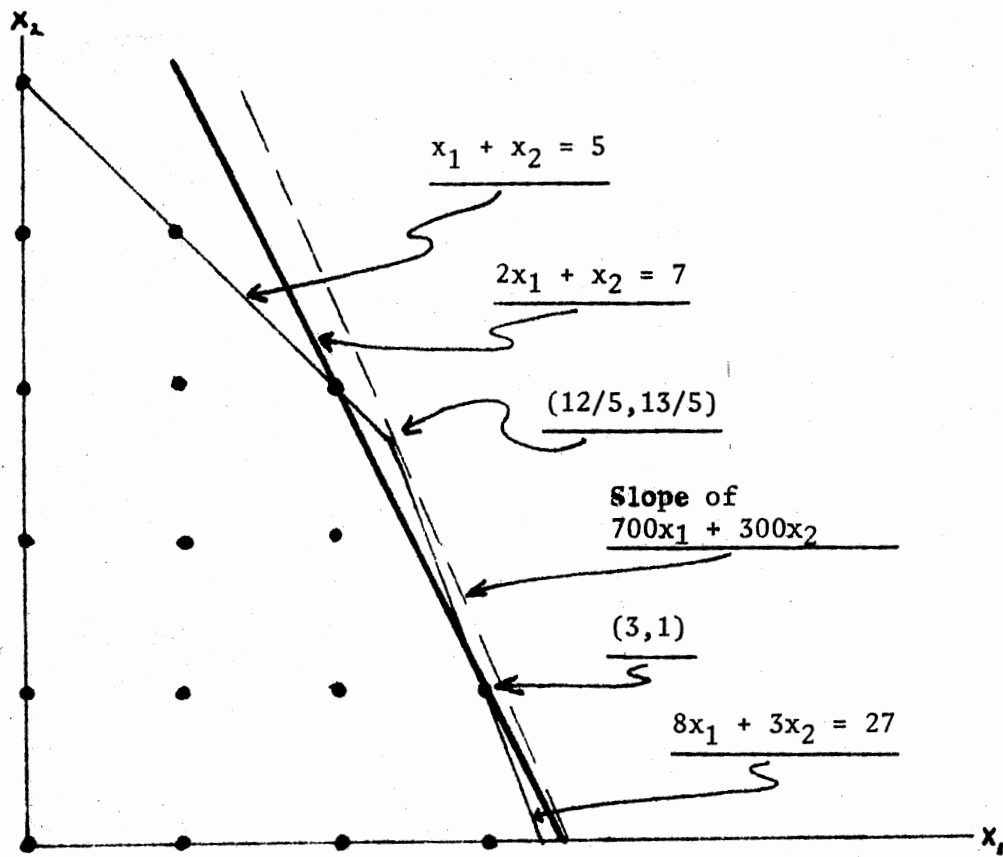


Figure 4. Graph of Example 1.1.1 with a Valid Cut

are coincident with a face of the convex hull of the ILP-feasible set, and therefore cut as deeply as possible into the feasible set without cutting off an ILP-feasible vector.

It may be noted that (3, 1), the optimal solution found for the ILP of Example 1.1.1, is not the closest ILP-feasible vector to the initial LP optimal solution, (12/5, 13/5). In fact there are five ILP-feasible vectors closer to (12/5, 13/5) than is (3, 1). This situation is not untypical and demonstrates the need for techniques designed specifically for the solution of the ILP. A. Story and H. Wagner (41) list several examples of scheduling problems solved by means of an ILP for which the ILP-feasible vector nearest the optimal solution of the associated LP gives an objective function value far from the optimal value of the ILP.

The Simplex Method

The discussion of cutting-plane methods disclosed the fact that a solution to the associated LP is required at possibly several points in the process of solving an ILP by means of a cut technique. This is also true of several of the enumeration methods. There exists an intimate relationship between integer and linear programming. For this reason the simplex method, the basic technique used to solve linear programs, will be reviewed.

Consider the associated LP of Example 1.1.1:

$$\begin{array}{ll}
 \text{maximize} & 700x_1 + 300x_2 \\
 \text{subject to} & 8x_1 + 3x_2 + x_3 = 27 \\
 & x_1 + x_2 + x_4 = 5, \text{ and } x_1, x_2 \geq 0.
 \end{array}$$

This canonical maximum linear program can be written in the form:

$$\text{maximize} \quad x_0 = c_B x_B + c_N x_N \quad (1.5.1)$$

$$\text{subject to} \quad Bx_B + Nx_N = b$$

$$\text{and} \quad x_B, x_N \geq 0,$$

$$\text{where } B = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 8 & 0 \\ 1 & 1 \end{bmatrix}, \quad x_B = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \quad x_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}, \quad c_B = \begin{bmatrix} 300 \\ 0 \end{bmatrix}, \quad c_N = \begin{bmatrix} 700 \\ 0 \end{bmatrix},$$

$$\text{and} \quad b = \begin{bmatrix} 27 \\ 5 \end{bmatrix}.$$

Since B^{-1} exists and equals $\begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix}$, a solution to (1.5.1) can be found by setting $x_N = 0$ and then

$$x_B = B^{-1}b = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 27 \\ 5 \end{bmatrix} = \begin{bmatrix} 5 \\ 12 \end{bmatrix}.$$

The vector $x = (x_B; x_N) = (x_2, x_3; x_1, x_4) = (5, 12; 0, 0)$ is called a basic solution to the linear program, and the matrix

$$B = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}$$

is called a basis matrix to the LP. In general, when the columns of any canonical LP in the form $Ax = b$ are arranged so that $A = (B, N)$ where B is an $m \times m$, non-singular matrix, then B is called a basis matrix for the LP. A solution $x = (x_B, x_N)$, where $x_B = B^{-1}b$ and $x_N = 0$, is called a basic solution. If the basic solution is also in the feasible set it is called a basic feasible solution. The variables of the vector x_B are those associated with the columns of the basis matrix B , and are therefore called basic variables. Likewise, the variables of the vector x_N associated with the columns of N are called nonbasic variables.

A fundamental result in linear programming is the following: If a LP has an optimal solution, it has a basic optimal solution. (See for example D. Gale [18, p.84]). This implies that for a bounded LP, for

which an optimal solution exists, a basic feasible solution exists which is optimal. Thus, at least one basic feasible solution exists along with its basis matrix B . In most cases more than one basis matrix can be found. In fact it is possible that up to $\binom{n}{m}$ different basis matrices could be found since each basis matrix is made up of n columns of A taken m at a time.

Table 1 gives the six basis matrices for the LP of Example 1.1.1 along with the inverse of each and the value of x_B of the corresponding basic solution. Figure 5 represents a graph of the feasible set of the LP and the x_1, x_2 -coordinates of the six basic solutions. It can be seen that the basic feasible solutions on the graph correspond to extreme points $(0, 0)$, $(0, 5)$, $(12/5, 13/5)$, and $(27/8, 0)$ of the convex polytope representing the feasible set. In fact it can be shown that a feasible vector corresponds to an extreme point if and only if it is a basic solution (see D. Luenberger [36, p. 21]). In that light it can be seen that the previous statement that the existence of an optimal solution implies the existence of an optimal extreme point solution is the geometric equivalent of the statement that the existence of an optimal solution implies the existence of a basic optimal solution.

The plan of the simplex method, in geometric terms, is to find an extreme point of the convex polytope of the feasible set of the bounded, maximum LP, and then move from extreme point to adjacent extreme point increasing the value of the objective function with each move until an optimal solution is reached. The fact that the convex polytope has only a finite number of extreme points guarantees a local optimal solution will be reached in a finite number of moves.

The simplex method can also be described in algebraic terms. First a basic feasible solution is found; then an "adjacent" basic feasible

TABLE I
BASIS MATRICES AND BASIC SOLUTIONS FOR EXAMPLE 1.1.1

B	B^{-1}	x_B
$\begin{bmatrix} 8 & 3 \\ 1 & 1 \end{bmatrix} = B_1$	$\begin{bmatrix} 1/5 & -3/5 \\ -1/5 & 8/5 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 12/5 \\ 13/5 \end{bmatrix}$
$\begin{bmatrix} 8 & 1 \\ 1 & 0 \end{bmatrix} = B_2$	$\begin{bmatrix} 0 & 1 \\ 1 & -8 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -13 \end{bmatrix}$
$\begin{bmatrix} 8 & 0 \\ 1 & 1 \end{bmatrix} = B_3$	$\begin{bmatrix} 1/8 & 0 \\ -1/8 & 1 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 27/8 \\ 13/8 \end{bmatrix}$
$\begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} = B_4$	$\begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix}$	$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 12 \end{bmatrix}$
$\begin{bmatrix} 3 & 0 \\ 1 & 1 \end{bmatrix} = B_5$	$\begin{bmatrix} 1/3 & 0 \\ -1/3 & 1 \end{bmatrix}$	$\begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} 9 \\ -4 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = B_6$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 27 \\ 5 \end{bmatrix}$

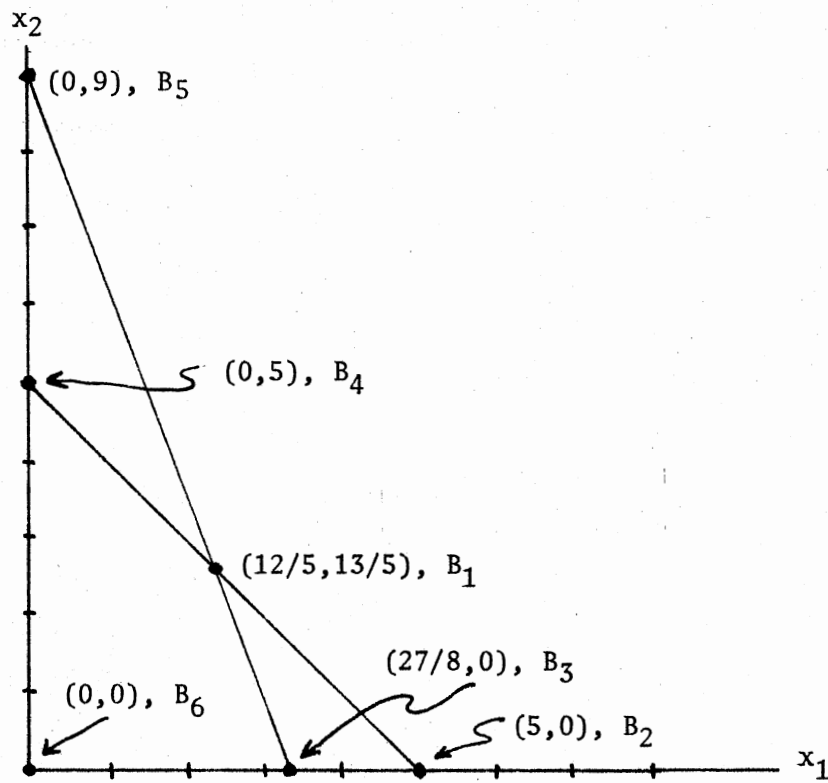


Figure 5. Basic Solutions of the Associated LP to Example 1.1.1 Plotted with x_1, x_2 -Coordinates

solution which increases the value of the objective function is found. This second basic feasible solution is adjacent to the first in the sense that it has the same basic variables but one. The process is continued until the objective function value cannot be improved by any adjacent basic feasible solution. The fact that there are no more than $\binom{n}{m}$ basic feasible solutions, together with the fact that improvement in the objective function value occurs at each iteration implies a local optimal solution will be found in a finite number of iterations.

A result from classical optimization (12), sometimes referred to as "The Fundamental Theorem of Convex Programming", states that any local maximum (or minimum) of a convex programming problem is a global maximum (or minimum). A convex programming problem is one in which a concave function is maximized over a convex set or a convex function is minimized over a convex set. Since every LP is either a maximization or a minimization of a linear function (which is both convex and concave) over a convex feasible set, then any local optimum found by the simplex method will be global. Thus, the simplex method will converge in a finite number of moves.

While it is true that for each extreme point of the convex polytope representing the feasible set of a bounded LP there exists a corresponding basic feasible solution, the correspondence is not one-to-one. There may be more than one basic feasible solution at a particular extreme point. This situation, described here in geometric terms, is referred to as degeneracy. A degenerate basic solution is one in which at least one basic variable has a value of zero. Thus, in moving from a degenerate basic feasible solution to an adjacent degenerate basic feasible solution, two basic variables with value zero may be interchanged leaving the objective function value unchanged. Consider the following example:

$$\begin{array}{ll}
 \text{Example 1.5.1} & \text{maximize} \quad 2x_1 + 5x_2 \\
 & \text{subject to} \quad x_1 + 2x_2 + x_3 = 2 \\
 & \quad \quad \quad 3x_1 + x_2 + x_4 = 6, \\
 & \quad \quad \quad x_1, x_2, x_3, x_4 \geq 0.
 \end{array}$$

The two degenerate basic feasible solutions,

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix},$$

both give a value of four to the objective function, and they both lie on the same extreme point of the feasible set, $(x_1, x_2) = (2, 0)$. The simplex method as described algebraically could possibly alternate between these two basic solutions without ever getting to a solution which would increase the value of the objective function.

What is needed is a method to prevent the selection of a basic solution with variables of the same value as those of the current basic solution. The method commonly used is based upon the concept of lexicographic ordering. A vector is lexicographically positive if its first non-zero entry is positive. A vector x is lexicographically greater than a vector y if the vector $x - y$ is lexicographically positive. The simplex procedure can be modified so that choices of departing variables depend upon lexicographic ordering of vectors thus preventing the interchange of two lexicographically equal degenerate vectors.

The simplex method can be written in terms of an algorithm. But before that can be done some notation must be introduced. Suppose that a canonical maximum LP:

$$\begin{array}{ll}
 \text{maximize} & xc \\
 \text{subject to} & Ax = b, x \geq 0,
 \end{array} \tag{1.5.2}$$

is to be solved. If the basis matrix, B , corresponding to a basic feasible

solution exists then the LP (1.5.2) can be written:

$$\text{maximize} \quad x_0 = c_B x_B + c_N x_N \quad (1.5.3)$$

$$\text{subject to} \quad Bx_B + Nx_N = b, \quad x_B \geq 0, \quad x_N \geq 0. \quad (1.5.4)$$

Since B^{-1} exists, (1.5.4) implies that $x_B = B^{-1}b - B^{-1}Nx_N$. Using this expression for x_B , (1.5.3) can be written $x_0 = c_B B^{-1}b - (c_B B^{-1}N - c_N)x_N$.

These results can be recorded in matrix form as:

$$\begin{bmatrix} x_0 \\ x_B \end{bmatrix} = \begin{bmatrix} c_B B^{-1}b \\ B^{-1}b \end{bmatrix} - \begin{bmatrix} c_B B^{-1}N - c_N \\ B^{-1}N \end{bmatrix} \cdot x_N. \quad (1.5.5)$$

To simplify this notation define

$$y_0 = \begin{bmatrix} c_B B^{-1}b \\ B^{-1}b \end{bmatrix} = \begin{bmatrix} y_{00} \\ y_{10} \\ \vdots \\ y_{m0} \end{bmatrix},$$

$$y_j = \begin{bmatrix} c_B B^{-1} a_j - c_j \\ B^{-1} a_j \end{bmatrix} = \begin{bmatrix} y_{0j} \\ y_{1j} \\ \vdots \\ y_{mj} \end{bmatrix}, \quad x_{B_0} = x_0, \quad (x_{B_1}, \dots, x_{B_m})^T = x_B,$$

a_j as the j^{th} column of N , and R as the index set for the columns of N .

With this notation (1.5.5) can be written:

$$\begin{bmatrix} x_{B_0} \\ x_{B_1} \\ \vdots \\ x_{B_r} \\ \vdots \\ x_{B_m} \end{bmatrix} = \begin{bmatrix} y_{00} \\ y_{10} \\ \vdots \\ y_{r0} \\ \vdots \\ y_{m0} \end{bmatrix} - \sum_{j \in R} \begin{bmatrix} y_{0j} \\ y_{1j} \\ \vdots \\ y_{rj} \\ \vdots \\ y_{mj} \end{bmatrix} \cdot x_j. \quad (1.5.6)$$

(1.5.6) contains all the necessary information for each basic feasible solution to (1.5.2). By setting $x_j = 0$ in (1.5.6) for all j in R (i.e., by setting $x_N = 0$) the value of the basic feasible solution corresponding

to the basis matrix B can be determined along with x_0 , the value of the objective function. When a move is made to an adjacent basic feasible solution equation (1.5.6) reflects this by replacing x_{B_r} (the departing variable) with x_k (the entering variable) in the matrix on its left-hand side.

Equation (1.5.6) is usually represented in the form of a table called a simplex tableau. Table II is a simplex tableau which represents (1.5.6) by assuming an equal sign after the column of basic variables, plus signs after each of the remaining columns (except the last), and that each column with a nonbasic variable above it is multiplied by that variable.

Given this notation, the simplex algorithm for finding an optimal solution for (1.5.2) by means of finding improved basic feasible solutions can be summarized:

STEP 1. (Initialization.) Find an initial basic feasible solution with basis matrix B . Go to STEP 2.

STEP 2. (Test for optimality.) If $y_{0j} \geq 0$ for every j in R then the current basic feasible solution is optimal. If $y_{0j} < 0$ for some j in R , go to STEP 3.

STEP 3. (Choosing an entering variable which will improve the objective function value.) Select any variable x_k , k in R , to enter the basis so long as $y_{0k} < 0$. Go to STEP 4.

STEP 4. (Choosing a departing variable so as to keep the basic solution feasible.) If $y_{ik} \leq 0$ for all $i = 1, \dots, m$ then no optimal solution exists because of the unboundedness of the feasible set. Otherwise, find minimum, $i = 1, \dots, m \{y_{i0}/y_{ik} \mid y_{ik} > 0\}$. If $i = r$ gives the minimum, choose x_{B_r} to leave the basis. Go to

TABLE II
THE INITIAL SIMPLEX TABLEAU

		Nonbasic Variables			
		...	$-x_j$...	\downarrow $-x_k$...
Basic Variables	x_0	y_{00}	y_{0j}	y_{0k}	
	
	
	
	x_{B_i}	y_{i0}	y_{ij}	y_{ik}	
	
	
	
	x_{B_r}	y_{r0}	y_{rj}	y_{rk}	
	
	x_{B_m}	y_{m0}	y_{mj}	y_{mk}	

TABLE III
THE SECOND SIMPLEX TABLEAU

		Nonbasic Variables			
		...	$-x_j$...	$-x_{B_r}$...
Basic Variables	x_0	$y_{00} - (y_{0k}y_{r0}/y_{rk})$	$y_{0j} - (y_{0k}y_{rj}/y_{rk})$	$-y_{0k}/y_{rk}$	
	
	
	
	x_{B_i}	$y_{i0} - (y_{ik}y_{r0}/y_{rk})$	$y_{ij} - (y_{ik}y_{rj}/y_{rk})$	$-y_{ik}/y_{rk}$	
	
	
	
	x_k	y_{r0} / y_{rk}	y_{rj} / y_{rk}	$1 / y_{rk}$	
	

STEP 5.

STEP 5. (Pivoting to rewrite the tableau in terms of the new set of basic variables.)

- (i) Divide the r^{th} row of the tableau by y_{rk} , the pivot element.
- (ii) Multiply the new r^{th} row by y_{ik} and subtract it from the i^{th} row for $i = 0, 1, \dots, m; i \neq r$.
- (iii) Replace the old k^{th} column by its negative divided by y_{rk} , except that y_{rk} itself is replaced by $1/y_{rk}$.

Go to STEP 2.

Table III represents Table II after one iteration of the simplex algorithm (i.e., one pass through steps 1 through 5).

For an example of the simplex algorithm applied to an LP recall the associated LP to Example 1.1.1. Choose as a basis matrix $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Therefore, x_3 and x_4 are basic variables, x_1 and x_2 are nonbasic, the initial basic feasible solution is $(x_B; x_N) = (x_3, x_4; x_1, x_2) = (27, 5; 0, 0)$, and the initial objective function value is $x_0 = 0$.

Figure 6 lists the initial tableau and succeeding tableaus for each iteration of the simplex algorithm. Arrows indicate the variables entering and departing from the basis. The pivot element is the element whose row and column both have arrows. The pivot element for the first iteration is $y_{rk} = y_{22} = 1$. Basic variables after the first iteration are x_3 and x_2 , the corresponding basic feasible solution is $(x_B; x_N) = (x_3, x_2; x_1, x_4) = (12, 5; 0, 0)$, and the objective function value is $x_0 = 1500$. Since $y_{01} = -400$ is negative, the optimal value has not been reached.

The pivot element for the second iteration is $y_{rk} = y_{11} = 5$. The basis variables are x_1 and x_2 after the second iteration, the corresponding

	$-x_1$	\downarrow $-x_2$	
x_0	0	-700	-300
x_3	27	8	3
$\leftarrow x_4$	5	1	1

Initial Tableau

	\downarrow $-x_1$	$-x_2$	
x_0	1500	-400	300
$\leftarrow x_3$	12	5	-3
x_2	5	1	1

Second Tableau

	$-x_3$	$-x_4$	
x_0	2460	80	60
x_1	12/5	1/5	-3/5
x_2	13/5	-1/5	8/5

Final Tableau

Figure 6. Tableaux for
the Simplex Algorithm
Applied to the LP of
Example 1.1.1

basic feasible solution is $(x_B; x_N) = (x_1, x_2; x_3, x_4) = (12/5, 13/5; 0, 0)$, and the objective function value is 2460. Since $y_{0j} \geq 0$ for $j = 3, 4$ this represents an optimal solution to the LP.

In terms of the graph of this LP in Figure 2, the simplex algorithm corresponds to beginning with the initial extreme point $(x_1, x_2) = (0, 0)$, moving (by means of the first iteration) to the adjacent extreme point $(x_1, x_2) = (0, 5)$, and finally to $(x_1, x_2) = (12/5, 13/5)$.

In the previous example an initial basic feasible solution $(x_B; x_N) = (x_3, x_4; x_1, x_2) = (27, 5; 0, 0)$ was apparent; but this may not always be the case. Fortunately, the simplex procedure itself can be used to locate an initial basic feasible solution. The problem is to find a basic non-negative solution of:

$$Ax = b. \quad (1.5.7)$$

This can be set up as a maximization problem:

$$\begin{aligned} &\text{maximize} && w(-u) && (1.5.8) \\ &\text{subject to} && \begin{bmatrix} A' & \vdots & I_m \end{bmatrix} \cdot \begin{bmatrix} x \\ w \end{bmatrix} = b', \quad x, w \geq 0, \end{aligned}$$

where w is an m -vector, u is the vector $(1, 1, \dots, 1)$ in R^m , and A' and b' correspond to A and b in (1.5.7), but with signs changed so that $b' \geq 0$. This problem (1.5.8) is called the corresponding problem to (1.5.7).

Now, (1.5.7) will have a non-negative solution if and only if the maximum value of (1.5.8) is zero, in which case the value of w will be zero and the value of x will solve (1.5.7), and thus be the desired initial basic feasible solution.

To illustrate this point consider the problem of finding an initial basic feasible solution to the LP of the following example:

Example 1.5.1

$$\begin{aligned}
&\text{maximize} && 2x_1 + x_2 \\
&\text{subject to} && -2x_1 + 2x_2 + 2x_3 = -4 \\
&&& 5x_1 + 3x_2 + 2x_4 = 15 \\
&\text{and} && x_1, x_2, x_3, x_4 \geq 0.
\end{aligned}$$

The corresponding problem (1.5.8) may be written:

$$\begin{aligned}
&\text{maximize} && (w_1, w_2) \cdot (-1, -1) \\
&\text{subject to} && \begin{bmatrix} 2 & 2 & -2 & 0 & 1 & 0 \\ 5 & 3 & 0 & 2 & 0 & 1 \end{bmatrix} \cdot (x_1, x_2, x_3, x_4, w_1, w_2)^T = \begin{bmatrix} 4 \\ 15 \end{bmatrix} \\
&\text{and} && x_1, x_2, x_3, x_4, w_1, w_2 \geq 0.
\end{aligned}$$

Since $b' = \begin{bmatrix} 4 \\ 15 \end{bmatrix}$ is non-negative, it is seen that taking $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for a basis matrix, and thus, w_1 and w_2 for basic variables, will result in a basic feasible solution for the corresponding problem. Therefore, this corresponding problem can be solved by the simplex algorithm yielding an optimal solution $(x_1, x_2, x_3, x_4, w_1, w_2) = (2, 0, 0, 5/2, 0, 0)$. Therefore, the vector $(x_1, x_2, x_3, x_4) = (2, 0, 0, 5/2)$ represents an initial basic feasible solution to the original LP.

The standard maximum LP:

$$\begin{aligned}
&\text{maximize} && xc \\
&\text{subject to} && Ax \leq b \\
&\text{and} && x \geq 0,
\end{aligned} \tag{1.5.9}$$

is related in a special way to the minimum LP:

$$\begin{aligned}
&\text{minimize} && vb \\
&\text{subject to} && vA \geq c \\
&\text{and} && v \geq 0.
\end{aligned} \tag{1.5.10}$$

The maximum LP (1.5.9) is called the primal problem and the minimum problem is called the dual problem.

The unique relationship between these two LP's is demonstrated by the Fundamental Theorem of Duality which was first noted by J. von Neumann (39).

If the primal and dual problems both have feasible solutions then they both have optimal solutions and their optimal objective function values are the same. If either problem has no feasible solution, then neither has an optimal solution.

As an illustration, consider the primal LP associated with Example 1.1.1:

$$\begin{array}{ll} \text{maximize} & (700, 300)x \\ \text{subject to} & \begin{bmatrix} 8 & 3 \\ 1 & 1 \end{bmatrix} x \leq \begin{bmatrix} 27 \\ 5 \end{bmatrix} \text{ and } x \geq 0. \end{array}$$

The corresponding dual problem is:

$$\begin{array}{ll} \text{minimize} & (27, 5)v \\ \text{subject to} & v \cdot \begin{bmatrix} 8 & 3 \\ 1 & 1 \end{bmatrix} \geq \begin{bmatrix} 700 \\ 300 \end{bmatrix} \text{ and } v \geq 0. \end{array}$$

If this dual problem is rewritten as a standard maximum LP and solved by the simplex procedure, an optimal solution $\bar{v} = (80, 60)$ is found. The resulting optimal objective function value is $(27, 5)\bar{v} = (27, 5)(80, 60) = 2460$. As indicated by the Fundamental Theorem of Duality this value is the same as the optimal objective function value of the primal problem: $(700, 300)\bar{x} = (700, 300)(12/5, 13/5) = 2460$.

It might be observed that the values 80 and 60 appear in the first row of the final tableau of Figure 6. This is no accident. Just as the current basic feasible solution of the primal problem can be determined from the first column of each tableau, a corresponding current basic solution to the dual problem can be determined from the first row. The initial tableau of Figure 6 implies an initial dual basic solution

$(v_3, v_4; v_1, v_2) = (-700, -300; 0, 0)$ with an objective function value of 0, and the second tableau implies a dual basic solution $(v_3, v_2; v_1, v_4) = (-400, 300, 0, 0)$ with an objective function value of 1500. Except for that of the final tableau, these basic solutions of the dual are not feasible and their objective function values are both less than the minimum of 2460.

These facts suggest an alternative way of viewing the simplex process. The simplex algorithm can be described as a procedure which searches for an optimal solution while maintaining feasibility for the primal, but the same process from the point of view of the dual problem is one of searching for a basic feasible solution while maintaining the optimality. Is it possible then to define an algorithm for the solution of the primal which begins with a basic solution which is not necessarily feasible, but does satisfy the optimality conditions of the simplex algorithm (i.e., $y_{0j} \geq 0$ for every j in R) and then searches for a feasible solution by bringing new variables into the basis?

Such a procedure has been developed by C. Lemke (34) and is named the dual simplex method. This method can be summarized in terms of the previously developed notation:

STEP 1. (Initialization.) Find a basis B such that $y_{0j} \geq 0$ for every j in R (i.e., the optimality conditions of the simplex algorithm are satisfied). Go to STEP 2.

STEP 2. (Testing for primal feasibility.) If $y_{i0} \geq 0$ for $i = 1, \dots, m$ then the current basic solution is feasible and therefore, since it also satisfies $y_{0j} \geq 0$, must be optimal. If not, go to STEP 3.

STEP 3. (Choosing a departing variable to "improve" feasibility.)

Select any variable x_{B_r} to leave the basis so long as $y_{r0} < 0$. Go to STEP 4.

STEP 4. (Choosing an entering variable to maintain optimality

conditions.) If $y_{rj} \geq 0$ for all j in R then no optimal solution

exists due to the primal problem being infeasible. Otherwise, find maximum $\{y_{0j}/y_{rj} \mid y_{rj} < 0\}$. Suppose $j = k$ gives the minimum. Then choose x_k to enter the basis. Go to STEP 5.

STEP 5. (Pivoting to rewrite the tableau in terms of the new set

of basic variables.) Pivot as in STEP 5 of the simplex algorithm.

Go to STEP 2.

As an illustration of the dual simplex method consider the following example:

Example 1.5.2:

$$\begin{aligned} &\text{minimize} && x_1 + x_2 \\ &\text{subject to} && 2x_1 + x_2 \geq 4 \\ &&& x_1 + 2x_2 \geq 5 \\ &\text{and} && x_1, x_2 \geq 0. \end{aligned}$$

This problem can be restated as a canonical maximum problem:

$$\begin{aligned} &\text{maximize} && -x_1 - x_2 \\ &\text{subject to} && -2x_1 - x_2 + x_3 = -4 \\ &&& -x_1 - 2x_2 + x_4 = -5 \\ &\text{and} && x_1, x_2 \geq 0. \end{aligned}$$

The initial basis matrix $B = I_2$ with basic variables x_3 and x_4 seems to be a straightforward choice. However, the corresponding basic solution $(x_3, x_4; x_1, x_2) = (-4, -5; 0, 0)$ is not feasible. Thus, the dual simplex

method may solve this problem more efficiently than the simplex method.

The initial dual simplex tableau is:

	\downarrow $-x_1$	$-x_2$
x_0	0	1
$\leftarrow x_3$	-4	-2
x_4	-5	-1

Note that $y_{01} = 1$ and $y_{02} = 1$ are both non-negative as required. Since $y_{10} = -4$ and $y_{20} = -5$ are both negative, either x_3 or x_4 could be chosen to be the departing variable. If x_3 is chosen, the entering variable must be x_1 since the maximum $\{1/(-2), 1/(-1)\} = 1/(-2)$ implying that the pivot element is $y_{11} = -2$. After pivoting the second tableau can be written:

	$-x_3$	\downarrow $-x_2$
x_0	-2	1/2
x_1	2	-1/2
$\leftarrow x_4$	-3	-1/2

The departing variable chosen is x_4 since y_{20} is negative, and the entering variable is x_2 since the maximum $\{(1/2)/(-1/2), (1/2)/(-3/2)\} = (1/2)/(-3/2)$. After pivoting on $y_{22} = -3/2$ the next tableau can be written:

	$-x_3$	$-x_4$
x_0	-3	1/3
x_1	1	-2/3
x_2	2	1/3

Since $y_{01} \geq 0$ and $y_{02} \geq 0$ this tableau represents an optimal solution $(x_1, x_2; x_3, x_4) = (1, 2; 0, 0)$ with objective function value of -3.

As in the example above the dual simplex method is useful when an initial basic matrix is not apparent. However, its chief use in integer programming is as a component in some of the cutting-plane algorithms.

CHAPTER II

THE DEVELOPMENT OF THE EARLY CUT TECHNIQUES

Relationship to Linear Programming

One relationship between integer linear programming and linear programming involves the use of the dual simplex method in certain cutting-plane algorithms. To demonstrate this use of the dual simplex method consider Example 1.1.1. Figure 4, a graph of this problem, shows the feasible set for the associated LP along with a cut (1.4.1). The final simplex tableau for this LP (Figure 6) gives an optimal solution $(x_1, x_2) = (12/5, 13/5)$. The addition of the constraint (1.4.1) to the LP cuts this optimal solution from the feasible set. Thus, the addition of the constraint (1.4.1) to the final tableau of Figure 6 produces a new LP, and an initial basic solution which satisfies the optimality conditions ($y_{0j} \geq 0$ for every j in R) but is not feasible:

$$\begin{array}{c|ccc}
 & & -x_3 & -x_4 \\
 \hline
 x_0 & 2460 & 80 & 60 \\
 x_1 & 12/5 & 1/5 & -3/5 \\
 x_2 & 13/5 & -1/5 & 8/5 \\
 s_1 & -2/5 & -1/5 & -2/5
 \end{array} \tag{2.1.1}$$

The last row of the tableau (2.1.1) is a representation of the constraint (1.4.1) in terms of the nonbasic variables x_3 and x_4 and a new slack variable s_1 . The constraint (1.4.1) can be written in the form

$$2x_1 + x_2 + s_1 = 7 \quad (2.1.2)$$

where $s_1 \geq 0$. The second and third rows of tableau (2.1.1) represent equations for x_1 and x_2 which can be substituted into (2.1.2) to give

$$s_1 = -2/5 + 1/5x_3 + 2/5x_4 \quad (2.1.3)$$

Equation (2.1.3) is then represented in the last row of the tableau.

The basic solution of tableau (2.1.1) suggests the use of the dual simplex method because it satisfies the two requisites for an initial basic solution of the dual method: the optimality conditions are satisfied and the solution is infeasible. After one dual simplex iteration, pivoting on the lower right-hand corner element, tableau (2.1.1) becomes:

$$\begin{array}{c|ccc} & -x_3 & -s_1 & \\ \hline x_0 & 2400 & 50 & 150 \\ x_1 & 3 & 1/2 & -3/2 \\ x_2 & 1 & -1 & 4 \\ x_4 & 1 & 1/2 & -5/2 \end{array} \quad (2.1.4)$$

The basic solution represented by (2.1.4), $(x_1, x_2, x_4; x_3, s_1) = (3, 1, 1; 0, 0)$, satisfies both optimality and feasibility requirements and is consequently an optimal solution to the new LP. Furthermore, since the solution is also integral it is an optimal solution of the ILP of Example 1.1.1.

The cut (1.4.1) which led to an optimal solution of the example was produced by inspecting the geometry of the ILP. In general this will not be possible. Thus a question central to this paper arises: Is it possible to generate cuts like (1.4.1) for any ILP, which will eventually lead to a solution of the problem? And, if so, how? The remaining sections of this chapter sketch the development of techniques which generate such cuts.

Before looking at special techniques to solve the ILP it is reasonable to ask whether it is possible to identify an ILP problem whose optimal solutions are identical to those of its associated LP. Such a problem could be solved by the simplex method without additional work. Certainly any ILP whose associated LP had a feasible set with integral extreme points (i.e., extreme points with all coefficients integral) would be a problem of this class. A. Hoffman and J. Kruskal (31) have given a characterization of convex polytopes with integral extreme points in terms of the constraint coefficient matrix. This characterization involves a property known as unimodularity. A square integer matrix is said to be unimodular if its determinant is unity. An $m \times n$ integer matrix is said to be totally unimodular if every square, non-singular submatrix is unimodular. Hoffman and Kruskal have shown the following:

Theorem 2.1.1. Every extreme point of the convex polytope

$$P = \{x \mid Ax \leq b \text{ and } x \geq 0\}$$

is integral if A is totally unimodular.

Thus, the constraint coefficient matrix of an ILP can be examined to determine whether or not special techniques will be necessary in order to obtain a solution. G. Dantzig (13) noted that a particular property of the constraint coefficient matrix of the transportation problem permits an essential reduction of the computational procedure. I. Heller and C. Tompkins (30) have shown that this particular property of the constraint coefficient matrix implies that the matrix is totally unimodular, and, moreover, that this property also holds for the constraint coefficient matrices of the transshipment, assignment, maximum flow and shortest path problems.

Therefore, a large collection of ILP problems can be solved without

resorting to enumeration or cutting-plane techniques. However, there remain many ILP problems whose solution can be found only by resorting to methods such as those discussed in this paper. Examples of particular types of such problems appear in a later chapter. The remainder of this chapter and the next consist of an historical development of the cutting-plane techniques used to solve such problems.

Early Uses of the Cutting-Plane

The idea of adding inequalities to a linear programming problem to progress toward an integer solution was first advanced in a 1954 paper by G. Dantzig, R. Fulkerson, and S. Johnson (16). This paper dealt with the solution of a large-scale (49 cities) traveling salesman problem. In a traveling salesman problem a linear objective function x_0 is to be minimized over a finite set F (the ILP-feasible set). The solution proposed can be summarized in the form of an algorithm.

Algorithm 2.2.1. (16)

STEP 1. Choose by guessing an x in F to optimize x_0 . Call it x' .

Go to STEP 2.

STEP 2. Formulate an LP with objective function x_0 and feasible set C' such that C' contains F . Go to STEP 3.

STEP 3. Determine by means of the simplex method an optimal vector x'' for the LP with feasible set C' . If $x_0(x') = x_0(x'')$ then x' is optimal in C' and a fortiori in F : stop. Otherwise, go to STEP 4.

STEP 4. If x'' is in F then x'' is optimal in F : stop. Otherwise go to STEP 5.

STEP 5. Find a linear constraint not satisfied by x'' but satisfied by each vector of F . This constraint used to cut off the non-integral optimum x'' is the first cutting-plane. Go to STEP 6.

STEP 6. Add the constraint of STEP 5 to C' giving a new LP feasible set C'' . Replace C' by C'' . Go to STEP 3.

Several comments can be made about this procedure. First, the feasible set C' is not necessarily the feasible set of the associated LP of the traveling salesman ILP. Second, the cutting-plane of STEP 5 is not generated automatically from the simplex tableau, but is derived from characteristics unique to the traveling salesman problem. Third, no proof is given that the algorithm will converge to a solution in a finite number of iterations. That is to say, it is not shown to be a finite algorithm. Finally, the significance of the procedure is its use of the additional LP constraint -- the cut.

The next use of cuts to solve an ILP was given in a 1957 paper of H. Markowitz and A. Manne (37). The authors emphasized that the paper presented no automatic algorithm for the solution of the ILP problem, but rather "a general approach susceptible to individual variations, depending upon the problem and the judgment of the user". The algorithm proposed by Markowitz and Manne is essentially the same as that of Dantzig, Fulkerson, and Johnson. The aspects in which the later method differs from the earlier foreshadow those of methods which follow it. One difference is that the method of Markowitz and Manne is intended to be a general algorithm, one which will work for any ILP. A second difference is that this method uses both the simplex and the dual simplex methods to optimize the LP. (Lemke's [34] dual simplex method had been introduced

in 1954.) Finally, this method generates cuts by examining the current, non-integral optimal solution in an unsystematic way rather than by using characteristics unique to the problem.

This trial and error method of generating cuts is the weak point of the algorithm. As Dantzig (14) noted in a 1957 survey paper on discrete programming, "The procedure would be straightforward except that the procedure for generating the complete set of additional constraints is not known."

The Cutting-Plane Methods of Gomory

In May of 1958 A. W. Tucker communicated to the American Mathematical Society an "Outline of an Algorithm for Integer Solutions to Linear Programs" by Ralph E. Gomory (24) of Princeton University. This paper announced an algorithm, referred to as Gomory's Method of Integer Forms, which was similar to that of Markowitz and Manne, but with three fundamental exceptions. First, Gomory's method would converge to a solution in a finite number of iterations. Second, it began by solving the associated LP, not just any LP whose feasible set included the ILP-feasible set. Finally, it provided a method of systematically generating cuts using the final simplex tableau of the LP.

Because Gomory's Method of Integer Forms is basic to nearly every later cutting-plane algorithm it will be developed in some detail. First, the algorithm will be described in detail, along with examples. Next, the basic cut will be derived from the geometry of the ILP and its associated LP. Finally, the conditions for a finite algorithm will be discussed. The following theorem will facilitate the discussion of Gomory's Method of Integer Forms and later techniques:

Theorem 2.3.1. (Gomory (26)) Given the ILP

$$\text{maximize } cx \quad (2.3.1)$$

$$\text{subject to } Ax = b, x \geq 0 \text{ and integer,}$$

and a representation (as in a simplex tableau) of the associated LP to

(2.3.1) given by

$$x_{B_i} = y_{i0} - \sum_{j \in R} y_{ij} x_j, \quad i = 0, 1, \dots, m \quad (2.3.2)$$

so that the basic solution determined by (2.3.2) is $x_{B_i} = y_{i0}$, $i = 0, 1, \dots, m$, $x_j = 0$, j in R ; then the inequality

$$\sum_{j \in R} ([h] y_{ij} - [hy_{ij}]) x_j \geq [h] y_{i0} - [hy_{i0}],$$

where h is any non-zero real number and $[d]$ means the greatest integer not greater than d , is a valid cut for (2.3.1).

Proof of Theorem 2.3.1. Multiplying (2.3.2) by h gives

$$hx_{B_i} + \sum_{j \in R} hy_{ij} x_j = hy_{i0} \quad (2.3.3)$$

and since $x \geq 0$ and $d \geq [d]$

$$[h] x_{B_i} + \sum_{j \in R} [hy_{ij}] x_j \leq hy_{i0}. \quad (2.3.4)$$

Since x is required to be integer in (2.3.1) the left-hand side of (2.3.4) must be integer so that

$$[h] x_{B_i} + \sum_{j \in R} [hy_{ij}] x_j \leq [hy_{i0}]. \quad (2.3.5)$$

Multiplying (2.3.2) by $[h]$ and subtracting (2.3.5) gives

$$\sum_{j \in R} ([h] y_{ij} - [hy_{ij}]) x_j \geq [h] y_{i0} - [hy_{i0}]. \quad (2.3.6)$$

Therefore, any (integral) solution x to (2.3.1) satisfies (2.3.6) making that constraint a valid cut for any non-zero value of h .

If h is set to 1, (2.3.6) yields $\sum_{j \in R} (y_{ij} - [y_{ij}]) x_j \geq y_{i0} - [y_{i0}]$.

If, in addition, fractions f_{ij} are introduced such that $y_{ij} = [y_{ij}] + f_{ij}$, then the cut (2.3.6) can be further reduced to the form

$$\sum_{j \in R} f_{ij} x_j \geq f_{i0}, \quad (2.3.7)$$

or in terms of a slack variable s ,

$$s = -f_{i0} + \sum_{j \in R} f_{ij} x_j, \quad s \geq 0, \quad (2.3.8)$$

Gomory's Method of Integer Forms can now be stated in algorithm form.

Algorithm 2.3.1. (Gomory [26])

STEP 1. Solve the associated LP to (2.3.1) using the simplex method, giving the following final simplex tableau:

$$\begin{array}{c|cccc} & & \dots & -x_j & \dots & -x_k & \dots \\ \hline x_0 & y_{00} & \dots & y_{0j} & \dots & y_{0k} & \dots \\ \cdot & \cdot & & \cdot & & \cdot & \\ \cdot & \cdot & & \cdot & & \cdot & \\ x_{B_i} & y_{i0} & & y_{ij} & & y_{ik} & \\ \cdot & \cdot & & \cdot & & \cdot & \\ \cdot & \cdot & & \cdot & & \cdot & \\ x_{B_m} & y_{m0} & & y_{mj} & & y_{mk} & \end{array} \quad (2.3.9)$$

Go to STEP 2.

STEP 2. If the solution of the LP is integral, it is an optimal solution to (2.3.1): stop. Otherwise, go to STEP 3.

STEP 3. Choose any row r of (2.3.9) with $f_{r0} > 0$ and append cut (2.3.8) with $i = r$ to tableau (2.3.9) giving the following tableau:

$$\begin{array}{c|cccc}
 & \dots & -x_j & \dots & -x_k & \dots \\
 \hline
 x_0 & y_{00} & \dots & y_{0j} & \dots & y_{0k} & \dots \\
 \vdots & \vdots & & \vdots & & \vdots & \\
 x_{B_i} & y_{i0} & \dots & y_{ij} & \dots & y_{ik} & \dots \\
 \vdots & \vdots & & \vdots & & \vdots & \\
 x_{B_m} & y_{m0} & \dots & y_{mj} & \dots & y_{mk} & \dots \\
 s & -f_{r0} & \dots & -f_{rj} & \dots & -f_{rk} & \dots
 \end{array} \quad (2.3.10)$$

Go to STEP 4.

STEP 4. Optimize (2.3.10) using the dual simplex method. Go to STEP 2.

Note that in STEP 3 a row was chosen with $f_{r0} > 0$. This was done so that the addition of the cut would cause the tableau (2.3.10) to represent an infeasible solution to the new LP. In other words, the optimal solution to the initial LP has been cut off from the present feasible set.

As an illustration consider Algorithm 2.3.1 applied to the ILP of Example 1.1.1.

Example 2.3.1. STEP 1. The final simplex tableau for the associated LP of Example 1.1.1 is :

		$-x_3$	$-x_4$
x_0	2460	80	60
x_1	12/5	1/5	-3/5
x_2	13/5	-1/5	8/5

This corresponds to tableau (2.3.9).

STEP 2. The LP optimal solution $(x_1, x_2; x_3, x_4) = (12/5, 13/5; 0, 0)$ is not integral so go to STEP 3.

STEP 3. Either row 1 or row 2 could be chosen to generate the cut.

Choose row 2. Therefore, $f_{20} = 13/5 - [13/5] = 3/5$, $f_{23} = -1/5 - [-1/5] = 4/5$, and $f_{24} = 8/5 - [8/5] = 3/5$. The cut (2.3.8) then becomes $s_1 = -3/5 + 4/5x_3 + 3/5x_4$. The tableau (2.3.10) becomes:

		$-x_3$	\downarrow $-x_4$
x_0	2460	80	60
x_1	12/5	1/5	-3/5
x_2	13/5	-1/5	8/5
s_1	-3/5	-4/5	-3/5

STEP 4. One iteration of the dual simplex method yields the tableau:

		$-x_3$	$-s_1$
x_0	2400	0	100
x_1	3	-3/5	-1
x_2	1	-7/3	8/3
x_4	1	4/3	-5/3

STEP 2. The solution $(x_1, x_2; x_3, x_4) = (3, 1; 0, 1)$ is integral and thus optimizes the ILP (2.3.1).

The cut of STEP 3 can be written, following the procedure used to generate equation (2.1.2), in terms of the variables x_1 and x_2 as

$$7x_1 + 3x_2 \leq 24. \quad (2.3.11)$$

Figure 7 shows how this constraint can be seen to cut off the optimal solution $(x_1, x_2) = (12/5, 13/5)$, but no ILP-feasible solution.

The cut for the Method of Integer Forms can be explained in terms of the geometry of the problem. A useful cut, such as the one above, is one which cuts off the old LP optimal solution while retaining all ILP-feasible vectors. Such a cut can be generated by producing a hyperplane

which (i) contains the current LP optimal solution, (ii) does not contain any interior point of the feasible set, and (iii) can be pushed into the feasible set parallel to itself and stopped before it cuts off any ILP-feasible vector. Figure 7 shows a hyperplane through the optimal solution $(12/5, 13/5)$ and a second hyperplane (the cut 2.3.11) which results from pushing the first hyperplane into the feasible set and stopping it before it cuts off an ILP-feasible vector. Any such hyperplane, one which satisfies (i) and (ii) above, is said to support the feasible set at the extreme point representing the LP optimal solution.

The problem then is to find an equation for a hyperplane, such as that of Figure 7, which supports the feasible set at the current LP optimal solution and contains some type of indicator which can signal its being pushed into an ILP-feasible vector. Furthermore, such an equation must be obtainable from the LP tableau.

Consider the equation from row 1 of the final simplex tableau for Example 1.1.1:

$$x_2 = 13/5 - 1/5(-x_3) + 8/5(-x_4). \quad (2.3.12)$$

From this equation determine another involving x_2 , x_3 , and x_4 which also holds at $(x_1, x_2) = (12/5, 13/5)$. One possibility is

$$x_2 + c = 13/5 + [-1/5] (-x_3) + [8/5] (-x_4) \quad (2.3.13)$$

where c takes on the proper non-negative value. Eliminate x_2 in equations (2.3.12) and (2.3.13) giving the equation

$$c = 4/5x_3 + 3/5x_4. \quad (2.3.14)$$

Note that $c = 0$ when $x_3 = x_4 = 0$. Equation (2.3.14) can be written in terms of x_1 and x_2 as

$$7x_1 + 3x_2 = 123/5 - c. \quad (2.3.15)$$

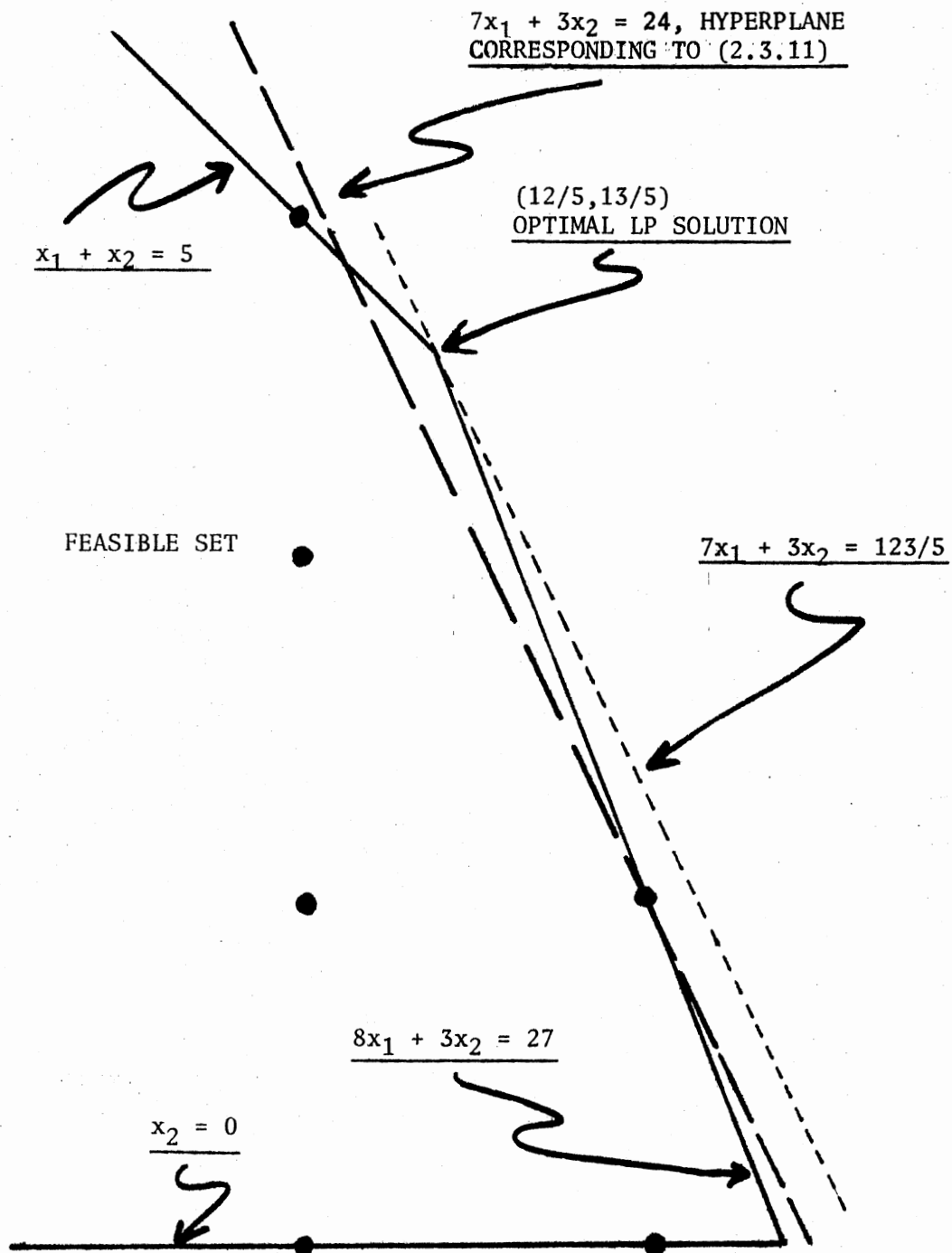


Figure 7. Feasible Set and Hyperplanes (Cuts) for Example 2.3.1

The expression on the left-hand side of (2.3.15) is a linear integer form in the variables x_1 and x_2 . When $c = 0$, as in the optimal solution $(x_1, x_2; x_3, x_4) = (12/5, 13/5; 0, 0)$, (2.3.15) represents a hyperplane which supports the feasible set at this optimal solution. As the value of c increases the hyperplane (2.3.15) moves into the feasible set.

The right-hand side of (2.3.15) does not become integral until $c = 3/5$. Therefore, because the left-hand side is an integer form, no lattice point will be encountered before $c = 3/5$. In other words the hyperplane (2.3.15) can be pushed toward the origin until its right-hand side becomes integral. The cut corresponding to $c = 3/5$ in (2.3.14) is

$$4/5x_3 + 3/5x_4 \geq 3/5.$$

This is the same as that found by means of (2.3.7).

In general the cut (2.3.7) will correspond to a hyperplane

$$\sum_{j \in R} f_{ij} x_j = 0 \quad (2.3.16)$$

satisfying the three conditions listed above. The hyperplane (2.3.16) will pass through the current optimal solution (i), since at that point $x_j = 0$ for all j in R . The hyperplane (2.3.16) is a supporting hyperplane at the current optimal extreme point solution (ii). For any vector interior to the feasible set all variables (slack and otherwise) must be positive. Therefore, all nonbasic variables, x_j , j in R , will have positive values at such an interior point. But since f_{ij_0} is positive for some j_0 in R (otherwise $f_{ij} = 0$ for all j in R and (2.3.16) fails to represent a hyperplane) no such interior point can satisfy (2.3.16). The integrality indicator (iii) is the integer form of the left-hand side of the equation which results from equation (2.3.16) being written in terms of the basic variables. Gomory (26) has shown that such an integer form

will always arise from (2.3.16). As in Example 2.3.1, if the right-hand side of (2.3.16) is increased from zero the first possible value to give the integer form an integer value is precisely f_{10} , giving the cut of (2.3.7).

It should be noted that an integer value for this integer form is necessary but not sufficient indication that the hyperplane has reached an ILP-feasible point. It may stop short of any such vector, but it will never cut one off.

Gomory (26) has shown that the Method of Integer Forms will find an optimal solution in a finite number of iterations provided STEP 3 is modified so that:

(1) the choice of row from tableau (2.3.9) maintains the columns of tableau (2.3.10) as lexicographically positive (i.e., the first non-zero component is positive);

(2) the row chosen is the topmost row with $f_{10} > 0$;

(3) any row which corresponds to a slack variable from a cut and had become non-basic is deleted on becoming basic again.

The finiteness of this algorithm is of course critical in terms of its utility. The Method of Integer Forms uses a type of cut which insures that the method will be finite. The same algorithm, but with a different type cut may not prove finite.

In June of 1958, shortly after Gomory had announced his Method of Integer Forms, a paper by George Dantzig (15) appeared. This paper presented a different type of cut for Gomory's algorithm. As Dantzig stated in the paper, "The procedure given here is not the same as Gomory's for adding additional constraints; indeed it is known to be weaker. It is worth noting because it is easy to see why the relationship holds and

because the relationship has a simple form."

Dantzig's replacement for the cut (2.3.8) was simply

$$\sum_{j \in R} x_j \geq 1, \text{ or}$$

$$s = -1 + \sum_{j \in R} x_j. \quad (2.3.17)$$

Using (2.3.17) in place of (2.3.8) in the algorithm (2.3.1) for the solution of Example (2.3.1) gives the following example:

Example 2.3.2.

STEP 1. The final LP tableau:

		$-x_3$	$-x_4$	
x_0	2460	80	60	
x_1	12/5	1/5	-3/5	
x_2	13/5	-1/5	8/5	,

STEP 2. This solution is not integral. Thus:

STEP 3. Choose row 2 to generate the cut. The cut (2.3.17) becomes

$$x_3 + x_4 \geq 1 \text{ or } s_1 = -1 + x_3 + x_4.$$

The tableau (2.3.10) becomes:

		$-x_3$	$-x_4$	
x_0	2460	80	60	
x_1	12/5	1/5	-3/5	
x_2	13/5	-1/5	8/5	
s_1	-1	-1	-1	

STEP 4. One dual simplex iteration of the above tableau yields:

		$-x_3$	$-x_4$
x_0	2400	20	60
x_1	3	$4/5$	$-3/5$
x_2	1	$-9/5$	$8/5$
x_4	1	1	-1

STEP 2. This solution is integral and thus optimizes the ILP of Example (2.3.1).

The cut STEP 3 can be written in terms of the variables x_1 and x_2 in the following way:

$$9x_1 + 4x_2 \leq 31.$$

Dantzig explains how the cut (2.3.17) is useful and valid. His reasoning, adapted to the ILP of Example (2.3.2), follows. Since x_3 and x_4 are both zero in the current LP solution $(x_1, x_2; x_3, x_4) = (12/5, 13/5; 0, 0)$ then this extreme point obviously cannot satisfy $x_3 + x_4 \geq 1$. Any non-degenerate solution of the associated LP other than $(12/5, 13/5; 0, 0)$ must have either x_3 or x_4 basic and therefore positive so that $x_3 + x_4 > 0$. But if the solution is required to be integral x_3 and x_4 must be integer valued so that $x_3 + x_4 \geq 1$ must be satisfied.

Ralph Gomory and H. Hoffman (28) show that this cut of Dantzig's will not always make Algorithm (2.3.1) finite. To see this consider the following example:

Example 2.3.3.

$$\begin{array}{ll}
 \text{maximize} & x_1 + x_2 \\
 \text{subject to} & 2x_1 + x_3 = 3 \\
 & 2x_2 + x_4 = 3 \\
 & x_1, x_2, x_3, x_4 \geq 0, x_1 \text{ and } x_2 \text{ integer.}
 \end{array}$$

Let $x = (x'_1, x'_2, x'_3, x'_4)$ be any non-degenerate basic feasible solution to the associated LP of the above ILP. Let $x'' = \text{minimum } \{x'_1, x'_2, x'_3, x'_4\}$. Let s_k denote the slack variable associated with the k^{th} added constraint of the form (2.3.17). If $x'' \geq 1$, then $s_k \geq x''$ for all k . This follows from the fact that if x'_p and x'_q are two nonbasic variables associated with the solution x then the cut (2.3.17) becomes $x'_p + x'_q \geq 1$ or $s_1 = x'_p + x'_q - 1$. But since x'' is no greater than either x'_p or x'_q and $x'' \geq 1$, $s_1 = x'_p + x'_q - 1 \geq 2x'' - 1 = x'' + (x'' - 1) \geq x''$. A similar argument shows that $s_k \geq x''$ for any k .

Now if $x = (x'_1, x'_2, x'_3, x'_4, s_1, s_2, \dots, s_k)$ is an optimal solution to the above ILP then two of these $k + 4$ variables must be nonbasic and therefore zero. But if x'' , the smallest of x'_1, x'_2, x'_3, x'_4 , satisfies $x'' \geq 1$, then all the slack variables generated by Dantzig cuts along with the three other original variables must be no smaller than x'' and thus none can be zero. The optimal solution for the above ILP can be seen by inspection to be $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$ so that $x'' \geq 1$, and therefore no non-degenerate, basic feasible solution generated by a Dantzig cut for this problem can be optimal.

In 1960 Gomory (25) developed two cutting-plane methods distinct from the Method of Integer Forms. The first of these was a method for the mixed integer linear program. A mixed integer linear program is a linear programming problem for which some but not all of the variables are constrained to be integer. This type of programming problem is, in a sense, a generalization of the ILP. It can be thought of as an LP tacked onto an ILP. Gomory derived a generalization of the cut (2.3.8) which when used in the algorithm of the Method of Integer Forms (Algorithm 2.3.1) in its place solves the mixed integer linear program.

The second of these cutting-plane methods is called the All-Integer Method. It was first described in the paper (27). This method like the Method of Integer Forms is an algorithm for the solution of the ILP. Its chief advantage over the earlier method lies in its exclusive use of integer entries in the tableaux and thus the elimination of rounding errors in the machine computation of the problem.

If an initial solution such as that of Table II is all-integer the succeeding tableau will also be all-integer if the pivot element y_{rk} has magnitude one. The All-Integer Method uses this fact in the following way. An initial solution is found which satisfies the optimality conditions (i.e., $y_{0j} \geq 0$ for j in R in the initial tableau) and is all-integer. If this solution is also feasible (i.e., $y_{i0} \geq 0$ for every i) then it must be an optimal solution to the ILP. If not, a dual simplex iteration is performed yielding a new solution closer to feasibility (i.e., a smaller objective function value). If the pivot element of this iteration is $y_{rk} = \pm 1$ the tableau will remain integral forcing the new solution to be integral. But the only way a pivot element of $y_{rk} = \pm 1$ can be guaranteed is by adding a row containing the necessary pivot element.

Such a row can be generated from the constraint (2.3.6) by choosing a proper value for h . If h is chosen so that $0 < h < 1$ then (2.3.6) becomes

$$\sum_{j \in R} -[hy_{rj}] x_j \geq -[hy_{r0}], \text{ or}$$

$$s + \sum_{j \in R} [hy_{rj}] x_j = [hy_{r0}]. \quad (2.3.18)$$

In general h must be chosen so that some $[hy_{rj}]$ of (2.3.18) has magnitude one and satisfies the conditions for a pivot element in the dual simplex method. Gomory (27) has proved the following theorem concerning the

choice of h and its relationship to the final simplex tableau of the associated LP.

Theorem 2.3.2. If the following inequality holds

$$h \leq \text{minimum} \{ (-1/y_{rj}) [y_{0j}/y_{0k}] \mid j \in R_r \}, \quad (2.3.19)$$

where k is the pivot column and $R_r = \{j \mid y_{rj} < 0, j \in R\}$, then $0 < h < 1$, $[hy_{rk}] = -1$, and $[hy_{rk}]$ satisfies the conditions for a dual simplex pivot element (i.e., y_{r0} is negative and $y_{0k} = \text{minimum} \{y_{0j} \mid j \in R_r\}$).

These ideas lead to the following algorithm for the All-Integer Method.

Algorithm 2.3.2. (Gomory [27])

STEP 1. Begin with a tableau representing a basic solution

$$\begin{array}{ccccccc} & & & \dots & -x_j & \dots & -x_k & \dots & \\ & & & & & & & & \\ x_0 & y_{00} & & \dots & y_{0j} & \dots & y_{0k} & \dots & \\ \vdots & \vdots & & & \vdots & & \vdots & & \\ \vdots & \vdots & & & \vdots & & \vdots & & \\ x_{B_r} & y_{r0} & & \dots & y_{rj} & \dots & y_{rk} & \dots & \\ \vdots & \vdots & & & \vdots & & \vdots & & \\ \vdots & \vdots & & & \vdots & & \vdots & & \\ x_{B_m} & y_{m0} & & & y_{mj} & & y_{mk} & & \end{array} \quad (2.3.20)$$

The tableau must also be all-integer (y_{ij} integral for $i = 0, 1, \dots, m$; $j \in R$) and satisfy the optimality conditions ($y_{0j} \geq 0$, for $j \in R$).

STEP 2. If the solution in STEP 1 is feasible (i.e., $y_{i0} \geq 0$, $i = 1, \dots, m$) then it is an optimal solution to the ILP. If not, go to STEP 3.

STEP 3. Choose a row $i = r \neq 0$ such that $y_{r0} < 0$, and a column

$j = k \neq 0$ such that $y_{0k} = \text{minimum}_{j \in R_r} y_{0j}$. Compute h according to (2.3.19). Add the cut (2.3.18) with the computed value of h to the tableau (2.3.20)

giving the following tableau:

		$\dots -x_j \dots -x_k \dots$	
x_0	y_{00}	$\dots y_{0j} \dots y_{0k} \dots$	
	.	.	.
	.	.	.
x_{B_r}	y_{r0}	$\dots y_{rj} \dots y_{rk} \dots$	
	.	.	.
	.	.	.
x_{B_m}	y_{m0}	$\dots y_{mj} \dots y_{mk} \dots$	
$+s_1$	$[hy_{r0}]$	$\dots [hy_{rj}] \dots [hy_{rk}] \dots$	

STEP 4. Execute a dual simplex iteration, pivoting on the element in the row of s_1 and the column k . The value of h in (2.3.18) was chosen so that this pivot element would have a value -1 . This will produce a new basic solution. Go to STEP 2.

Note: if the initial basic solution does not satisfy the optimality conditions a superfluous constraint of the form $s_0 = M - \sum_{j \in R} x_j$ can be appended to the tableau, and a dual simplex iteration performed. This constraint with M an arbitrarily large integer constant will not cut into the feasible set, but will provide a pivot element with value of 1. Such a constraint forces the sum of the nonbasic variables ($\sum_{j \in R} x_j$) to be large and positive. Since the top row of the tableau ($x_0 = y_{00} - \sum_{j \in R} y_{0j} x_j$) represents an equation for the objective function value which must decrease in a dual simplex iteration, a large value of $\sum_{j \in R} x_j$ would tend to eliminate negative values of the y_{0j} , thus producing a solution satisfying the optimality conditions.

Consider Algorithm 2.3.2 applied to the ILP of Example 1.1.1.

Example 2.3.4.

STEP 1. The initial tableau for this ILP is

		$-x_1$	$-x_2$
x_0	0	-700	-300
x_3	27	8	3
x_4	5	1	1

Since this tableau represents a solution which does not satisfy the optimality conditions, a constraint of the form $s_0 = M - (x_1 + x_2)$ could be appended. But if 5 is a large enough value for M, such a constraint already exists in the last row of the present tableau. Pivoting on element $y_{21} = 1$, bringing x_1 into the basis and removing x_4 gives the following tableau representing a basic solution, as in tableau (2.3.20):

		$-x_4$	$-x_2$
x_0	3500	700	400
x_3	-13	-8	-5
x_1	5	1	1

STEP 2. Since $x_3 = -13$ the previous tableau represents an infeasible solution. Go to STEP 3.

STEP 3. Choose row 3 since $y_{10} = -13 < 0$ and column 2 since $y_{02} = \text{minimum } \{700, 400\}$. Compute $h \leq \text{minimum } \{ (-1/-8) [700/400], (-1/-5) [400/400] \} = 1/8$. Add the cut from (2.3.18):

$$s_1 + [-8/8] x_4 + [-5/8] x_2 = [-13/8] ,$$

to the last tableau of STEP 1 giving the following tableau:

		$-x_4$	\downarrow $-x_2$
x_0	3500	700	400
x_3	-13	-8	-5
x_1	5	1	1
$\leftarrow s_1$	-2	-1	-1

STEP 4. Pivoting on the element -1 of row s_1 and column 2 yields:

		$-x_4$	$-s_1$
x_0	2700	300	400
x_3	-3	-3	-5
x_1	3	0	1
x_2	2	1	-1

STEP 2. The solution of STEP 4 is not feasible. Go to STEP 3.

STEP 3. Choose row 3 again since $y_{10} = -3 < 0$ and column 4 since $y_{04} = \text{minimum } \{300, 400\}$. Compute $h \leq \text{minimum } \{(-1/-3) [300/300], (-1/-5)[400/300]\} = 1/5$. Add the cut from (2.3.18):

$$s_2 + [-3/5] x_4 + [-5/5] s_1 = [-3/5],$$

to the tableau of STEP 4 giving:

		\downarrow $-x_4$	$-s_1$
x_0	2700	300	400
x_3	-3	-3	-5
x_1	3	0	1
x_2	2	1	-1
$\leftarrow s_2$	-1	-1	-1

STEP 4. Pivoting on the element -1 of row s_2 column x_4 yields:

		$-s_2$	$-s_1$
x_0	2400	300	100
x_3	0	-3	-2
x_1	3	0	1
x_2	1	1	-2
x_4	1	-1	1

This tableau represents a basic solution $(x_1, x_2, x_3, x_4; s_1, s_2) = (3, 1, 0, 1; 0, 0)$.

STEP 2. Since the solution of STEP 4 is feasible, satisfies the optimality requirements, and is integral it must be an optimal solution to the ILP.

The geometry of the cut (2.3.18) is similar to that of the cut (2.2.8) of the Method of Integer Forms. An equation is developed for a hyperplane passing through the current basic solution. This hyperplane is pushed toward the origin until it begins to cut into the feasible set, and stops upon encountering a lattice point after cutting off part of the feasible set. This hyperplane passing through the first basic (infeasible) solution of STEP 1 of Example 2.3.4, $(x_1, x_2; x_3, x_4) = (5, 0; -13, 0)$, can be developed by beginning with the source row

$$x_3 = -13 + (-8)(-x_4) + (-5)(-x_2).$$

Multiplying by $h \neq 0$ gives

$$hx_3 = -13h + (-8h)(-x_4) + (-5h)(-x_2). \quad (2.3.21)$$

Since x_4 and x_2 are zero at the current solution, equation (2.3.21) with any coefficient substituted for $(-8h)$ and $(-5h)$ will yield an equation for a hyperplane passing through the current basic solution $(x_1, x_2; x_3, x_4) = (5, 0; -13, 0)$. The equation

$$hx_3 = -13h + [-8h](-x_4) + [-5h](-x_2) \quad (2.3.22)$$

is therefore the equation of a hyperplane passing through the current basic solution which can be easily derived from the final LP tableau.

However, the coefficient of x_3 might not be an integer. Since $[h] \leq h$ and $x_3 \leq 0$ before feasibility is reached, $[h]x_3 \geq hx_3$ or $[h]x_3 - c = hx_3$, where $c \geq 0$ when $x_3 \leq 0$. So that from (2.3.22):

$$[h]x_3 - c = -13h + [8h](-x_4) + [-5h](-x_2). \quad (2.3.23)$$

If $0 < h < 1$, as is the case in the All-Integer Method, equation (2.3.23) can be written

$$c - 13h = [-8h]x_4 + [-5h]x_2, \quad (2.3.24)$$

which can be written in terms of the variables x_1 and x_2 as

$$c - 13h - 5[-8h] = -[-8h]x_1 + ([-5h] - [-8h])x_2. \quad (2.3.25)$$

For $h = 1/8$, equation (2.3.25) becomes

$$c - 13/8 + 5 = x_1 \quad (2.3.26)$$

When $c = 13/8$ this hyperplane (2.3.26) passes through the current basic solution as can be seen in Figure 8. When $c = 0$ (2.3.26) represents a supporting hyperplane to the feasible set. And when $c = -3/8$ (2.3.26) will for the first time pass through an ILP-feasible vector. Note that when $c = 5/8$ the hyperplane (2.3.26) contains lattice points, but these are not ILP-feasible vectors since the hyperplane has not yet encountered the feasible set (at $c = 0$).

Thus, the first cut of Example 2.3.4 is generated from (2.3.26) when $c = -3/8$, that is, when (2.3.24) becomes $[-13h] = [-8h]x_4 + [-5h]x_2$, the equation of the hyperplane corresponding to the cut of (2.3.18). Figure 9 illustrates this same process generating the second cut of Example 2.3.4. A hyperplane is generated from equation (2.3.25) by choosing an appropriate value for h . This hyperplane passes through the current basic solution, $(x_1, x_2; x_3, x_4, s_1) = (3, 2; -3, 0, 0)$. The hyperplane is then

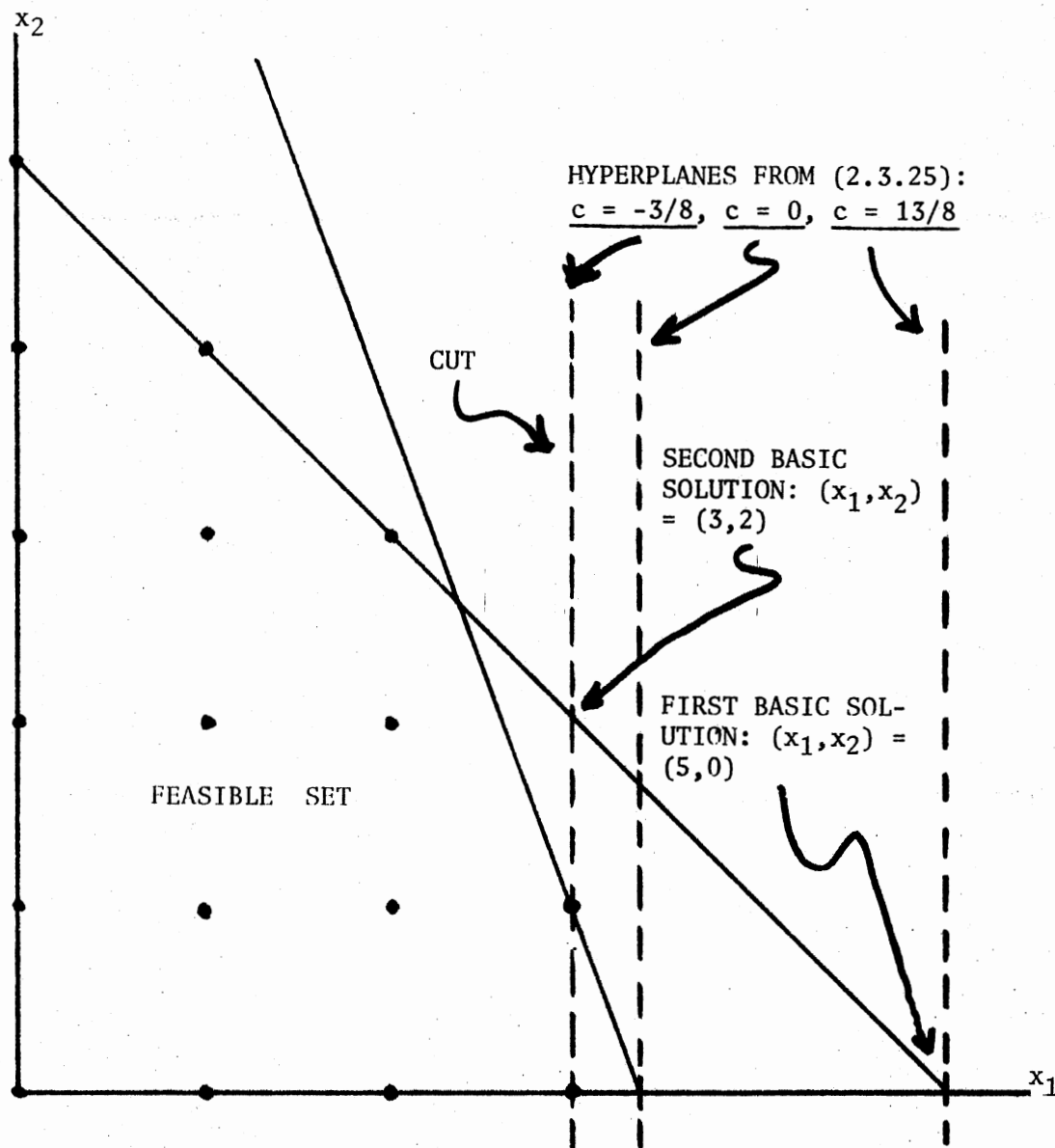


Figure 8. Basic Solutions and Cut-Generating Hyperplanes for Example 2.3.4, First Iteration

pushed toward the feasible set until it intercepts an extreme point (at $c = 0$). It is then pushed into the feasible set until the integer form comprising the right-hand side of (2.3.25) takes on an integer value, that is, until the hyperplane has (possibly) encountered a lattice point. At this point the hyperplane (2.3.25) with $c = -2/5$ corresponds to the second cut of Example 2.3.4: $2x_1 + x_2 \leq 7$, or in terms of the nonbasic variables: $s_2 = -1 + x_4 + s_1$.

As with the Method of Integer Forms the All-Integer Method is a finite algorithm when certain similar precautions are taken. While the All-Integer Method overcomes the problem of round-off error, it is prone to problems with degeneracy; and while degeneracy is itself overcome by the use of lexicographic ordering, it can often cause a marked increase in the number of iterations necessary for a solution (see, for example, C. Trauth and R. Woolsey [42]).

In a letter to the editor of Operations Research dated July 7, 1966, Robert Wilson (44) gave a modification of the All-Integer Method cut which produces "stronger" cuts than equation (2.3.18) with a value of h chosen by means of (2.3.19). The term "stronger" in relation to cutting-planes can be defined in the following way:

Definition 2.3.1. The cut

$$\sum_{j \in R} \pi'_j x_j \geq \pi'_0 \quad (2.3.27)$$

is a strengthening of the cut

$$\sum_{j \in R} \pi_j x_j \geq \pi_0 \quad (2.3.28)$$

if $\pi'_j \leq \pi_j$ for every j in R and $\pi'_0 \geq \pi_0$. Alternatively, the cut (2.3.28) is a weakening of (2.3.27).

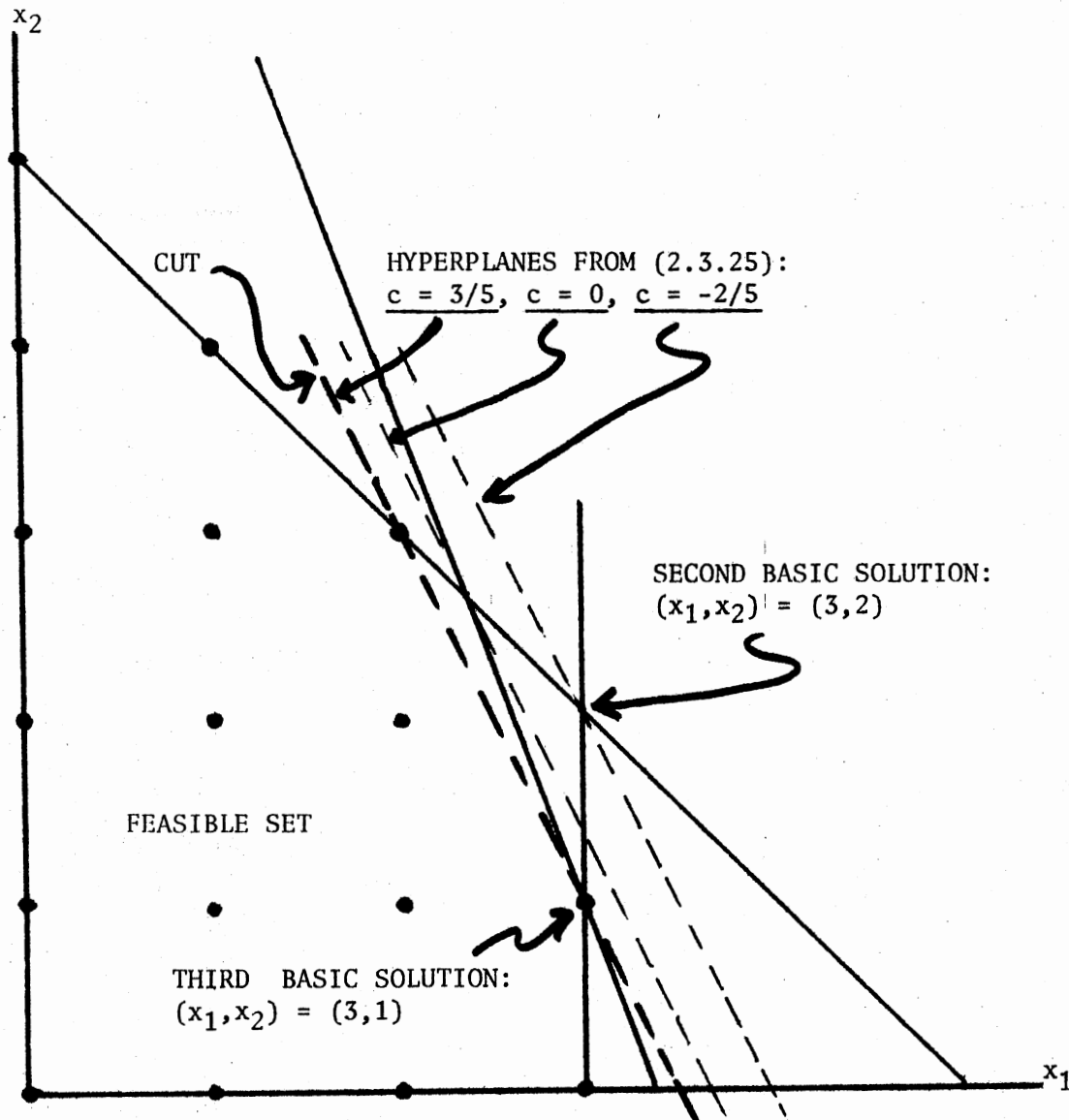


Figure 9. Basic Solutions and Cut-Generating Hyperplanes for Example 2.3.4, Second Iteration

If $x \geq 0$ satisfies (2.3.27) it will also satisfy (2.3.28) since $\pi_j x_j \geq \pi'_j x_j$ for all j in R implies $\sum_{j \in R} \pi_j x_j \geq \sum_{j \in R} \pi'_j x_j \geq \pi'_0 \geq \pi_0$. This shows that any feasible vector which satisfies a strengthening of a given cut will also satisfy the cut itself. For example, the cut (2.1.2) represented in terms of the variables x_3 and x_4 is $1/5x_3 + 2/5x_4 \geq 2/5$. In terms of definition 2.3.1 the cut $1/5x_3 + 1/5x_4 \geq 3/5$ is a strengthening of (2.1.2). This strengthened cut can be represented in terms of x_1 and x_2 as $9x_1 + 4x_2 \leq 29$. This strengthened cut lops off more of the feasible set than does (2.1.2). However, it also cuts off the ILP-feasible vectors $(x_1, x_2) = (3, 1)$ and $(x_1, x_2) = (2, 3)$ making it unusable for integer programming.

Wilson's refinement of the All-Integer Method can be demonstrated by means of an example:

Example 2.3.5. (R. Wilson): Consider an ILP problem with the following tableau corresponding to (2.3.20) of Algorithm 2.3.2

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	
x_0	20	1	2	3	4
x_5	-20	-7	-8	-15	18

Note that $r = 5$, $R = \{1, 2, 3, 4\}$, $R_r = R_5 = \{j \mid y_{5j} < 0, j \text{ in } R\} = \{1, 2, 3\}$. Column 1 is the pivot column since $1 = y_{01} = \text{minimum } \{y_{0j} \mid j \text{ is in } R_5\} = \text{minimum } \{1, 2, 3\}$. The All-Integer cut from (2.3.18) is then

$$s_1 + \sum_{j \in R} [hy_{5j}]x_j = [hy_{r0}], \text{ or in this case}$$

$$s_1 + [-7h]x_1 + [-8h]x_2 + [-15h]x_3 + [18h]x_4 = [-20h], \quad (2.3.29)$$

where h is defined as in (2.3.19). From (2.3.19) h can be calculated:

$$h = \text{minimum } \{(-1/y_{5j})[y_{0j}/y_{01}] \mid j \text{ is in } R_5\} = \text{minimum } \{1/7, 2/8, 3/15\} = 1/7.$$

With $h = 1/7$ the cut (2.3.29) becomes:

$$s_1 + [-7/7]x_1 + [-8/7]x_2 + [-15/7]x_3 + [18/7]x_4 = [-20/7], \quad (2.3.30)$$

or in simpler terms

$$x_1 + 2x_2 + 3x_3 - 2x_4 \geq 3. \quad (2.3.31)$$

Note that a smaller positive value of h will not affect the value of the coefficient of x_1 ; it may reduce the values of the coefficients of x_2 and x_3 and the constant; and it may increase the value of the coefficient of x_4 in (2.3.29). Any cut which serves as a strengthening of (2.3.31) must have smaller (or equal) values of the coefficients of x_1 , x_2 , x_3 , and x_4 and must have a larger (or equal) value for the constant. To accomplish this, the values h can assume are (term by term):

$$\text{for } x_1, x_2, x_3: \quad 0 \leq h \leq 1/7,$$

$$\text{for } x_4: \quad 1/9 \leq h \leq 1/7,$$

for the constant term: $1/(10 - \epsilon) \leq h \leq 1/7$, where $\epsilon > 0$ is small. Thus, $h^* = \text{maximum} \{1/(10 - \epsilon), 1/9\} = 1/9$ may produce a stronger cut for (2.3.29) than $h = 1/7$. In fact with $h^* = 1/9$, (2.3.29) yields

$$x_1 + x_2 + 2x_3 - 2x_4 \geq 3.$$

In general a choice for h in (2.3.18) which would yield a strengthened cut would be $h^* = \text{minimum} \{h, \text{maximum} \{h_j \mid j \text{ is not in } R_r\}\}$, where h is calculated as in (2.3.19), and $h_j = [hy_{rj}]/y_{rj}$, $j \geq 1$, $h_0 = (1 + [hy_{r0}])/(y_{r0} - \epsilon)$. If all $y_{rj} < 0$ so that $R = R_r$, then h^* may be taken arbitrarily close to zero and positive. Note that while the cut (2.3.18) with h^* substituted for h may be stronger, it will, unlike the cut of the example, remain valid since (2.3.18) is a special case of (2.3.6) which is a valid cut for any non-zero h .

CHAPTER III

THE DEVELOPMENT OF THE LATER CUT TECHNIQUES

Primal Methods

A major disadvantage of the methods of Chapter II lies in the fact that of all the basic solutions involved in the process only the final solution is ILP-feasible. Thus, if a machine implementation of the algorithm terminates prematurely the non-optimal solution returned is useless. If, however, an algorithm could be devised which proceeded from one ILP-feasible solution to another with each iteration improving the value of the objective function, then premature termination would produce at least an approximate to an optimal solution; and one which would be ILP-feasible.

In a 1962 Belgian paper A. Ben-Israel and A. Charnes (40) presented an algorithm they called a "Direct Algorithm" for the solution of the ILP. According to H. Salkin (40) this method involved cutting-planes and the movement from one ILP-feasible solution to another, but was very difficult to implement. Later, in a 1965 paper, R. Young (45) described a "Rudimentary Primal Algorithm" for the solution of the ILP. This method, different from that of Ben-Israel and Charnes in its details, was similar in that it used cutting-planes to move from one ILP-feasible to another. It was described as a "Primal" algorithm because it was based on Dantzig's (Primal) Simplex Method rather than the dual simplex method of the Gomory algorithms.

In 1967 F. Glover (21) published a paper giving a "pseudo-primal

algorithm" for solving ILP's. This algorithm represented a fundamentally different approach to a primal ILP method. Building upon this paper and Young's 1965 paper, Young (46) and Glover (22) brought out papers simultaneously in 1968 describing a simplified primal ILP algorithm. This algorithm in its basic form can be seen to be a direct modification of the simplex algorithm.

Algorithm 3.1.1. (Young and Glover)

STEP 1. Find an initial basic feasible integral solution. If one is not apparent a method similar to that of Example 1.5.1 may be used. The only adjustment to that method would be that the corresponding LP becomes a corresponding ILP. Go to STEP 2.

STEP 2. If $y_{0j} \geq 0$ for every j in R then the current basic feasible (integral) solution is optimal. If $y_{0j} < 0$ for some j in R , go to STEP 3.

STEP 3. Select any variable x_k , k in R , to enter the basis so long as $y_{0k} < 0$. Go to STEP 4.

STEP 4. If $y_{ik} \leq 0$ for all $i = 1, \dots, m$, then no optimal solution exists because of the unboundedness of the feasible set. Otherwise, find the minimum $\{y_{i0}/y_{ik} \mid y_{ik} \geq 1, i = 1, \dots, m\}$. Note that $y_{ik} \geq 0$ since the tableau consists of integer entries only. If $i = r$ gives the minimum and $y_{rk} = 1$, choose x_{B_r} to leave the basis. Go to STEP 6. If $i = r$ gives the minimum and $y_{rk} > 1$, go to STEP 5.

STEP 5. Append the cut

$$s + \sum_{j \in R} [y_{rj}/y_{rk}] x_j = [y_{r0}/y_{rk}] , \quad (3.1.1)$$

where $s \geq 0$. Choose s to leave the basis. Go to STEP 6.

STEP 6. Execute a simplex pivot as in STEP 5 of the simplex algorithm. Go to STEP 2.

The cut (3.1.1) is derived from the cut (2.3.18) by setting $h = 1/y_{rk}$. Then $0 < h < 1$ since $y_{rk} > 1$. Since x_k is the variable entering the basis and s is the departing variable, the pivot element of the tableau is $[y_{rk}/y_{rk}]$. Since this element has magnitude 1, any all-integer tableau will remain all-integer after pivoting.

To illustrate this algorithm consider the following example:

Example 3.1.1. (from Example 1.1.1)

maximize $700x_1 + 300x_2$

subject to $8x_1 + 3x_2 \leq 27$

$x_1 + x_2 \leq 5$, x_1 and $x_2 \geq 0$ and integer.

STEP 1. An initial basic feasible integer solution is obvious and can be seen in the initial tableau:

		$-x_1$	$-x_2$
x_0	0	-700	-300
x_3	27	8	3
x_4	5	1	1

STEP 2. Since $y_{01} = -700$ and $y_{02} = -300$ the current basic feasible (integral) solution is not optimal. Go to STEP 3.

STEP 3. Either x_1 or x_2 may come into the basis. Select x_1 .

STEP 4. The minimum $\{y_{i0}/y_{ik} \mid y_{ik} \geq 1, i = 1, \dots, m\} =$

minimum $\{y_{10}/y_{11}, y_{20}/y_{21}\} = \text{minimum } \{27/8, 5/1\} = 27/8$. Therefore,

$$y_{rk} = y_{11} = 8.$$

STEP 5. Append the cut:

$$s + \sum_{j \in R} [y_{rj}/y_{rk}]x_j = [y_{r0}/y_{rk}], \text{ that is}$$

$$s_1 + [y_{11}/y_{11}]x_1 + [y_{12}/y_{11}]x_2 = [y_{10}/y_{11}], \text{ or}$$

$$s_1 + x_1 + 0 = 3, \text{ to the initial tableau.}$$

This results in the tableau: (the appended constraint is in parenthesis)

		\downarrow $-x_1$	$-x_2$
x_0	0	-700	-300
x_3	27	8	3
x_4	5	1	1
s_1	3	1	0

$$(s_1 + x_1 + 0 = 3 \text{ or } x_1 \leq 3).$$

Choose s_1 to leave the basis.

STEP 6. Execute a simplex iteration bringing x_1 into the basis and taking s_1 out. Since the pivot element is 1, the new tableau will also be integral.

The following tableaux represent the results of iterations one through four. The appended cut constraint is shown in parenthesis in its derived form and also in terms of x_1 and x_2 alone.

SECOND TABLEAU

		$-s_1$	\downarrow $-x_2$
x_0	2100	700	-300
x_3	3	-8	3
x_4	2	-1	1
x_1	3	1	0
$\leftarrow s_2$	1	-3	1

$(s_2 - 3s_1 + x_2 = 1 \text{ or } 3x_1 + x_2 \leq 10).$

THIRD TABLEAU

		\downarrow $-s_1$	$-s_2$
x_0	2400	-200	300
$\leftarrow x_3$	0	1	-3
x_4	1	2	-1
x_1	3	1	0
x_2	1	-3	1

(No cut since the pivot is already 1).

FOURTH TABLEAU

		$-x_3$	\downarrow $-s_2$
x_0	2400	200	-300
s_1	0	1	-3
x_4	1	-2	5
x_1	3	-1	3
x_2	1	3	-8
$+s_3$	0	-1	1

$$(s_3 - x_3 + s_2 = 0 \text{ or } 5x_1 + 2x_2 \leq 17).$$

FIFTH TABLEAU

		\downarrow $-x_3$	$-s_3$
x_0	2400	-100	300
s_1	0	-2	3
x_4	1	3	-5
x_1	3	2	-3
x_2	1	-5	8
s_2	0	-1	1
$+s_4$	0	1	-2

$$(s_4 + x_3 - 2s_3 = 0 \text{ or } 2x_1 + x_2 \leq 7).$$

SIXTH TABLEAU

		$-s_4$	$-s_3$
x_0	2400	100	100
s_1	0	2	-1
x_4	1	-3	1
x_1	3	-2	1
x_2	1	5	-2
s_2	0	1	-1
x_3	0	1	-2

The third tableau represents an optimal solution. However, the optimality conditions are not satisfied until the sixth tableau. One interpretation of this situation is that the solutions represented in tableaux three through six are degenerate, and in each case a basic variable with value zero leaves the basis and is replaced by a variable with value zero. For example, in proceeding from the third to the fourth tableau x_3 (value zero) leaves the basis only to be replaced by s_1 (value zero). None of these exchanges causes a change in the objective function value since none of the zero-value variables contributed to its value in the first place.

An analysis of the geometry of this example helps to explain why the method of Young and Glover often requires additional iterations after an optimal solution is reached. As Figure 10 illustrates, the method runs through six different basic feasible solutions while encountering only three extreme points. The solution of the first tableau corresponds to the extreme point $(x_1, x_2) = (0, 0)$. Since x_1 is chosen to enter the basis (and become positive) the second basic feasible solution to the associated

LP would correspond to the extreme point $(x_1, x_2) = (27/8, 0)$. However, cut 1 ($x_1 \leq 3$) is appended to the associated LP before pivoting so as to force the second basic feasible solution to be integral. The value of x_1 will now increase from zero to 3 rather than from zero to $27/8$.

In general, if the cut (3.1.1) were modified so that the right-hand side were y_{r0}/y_{rk} instead of $\lfloor y_{r0}/y_{rk} \rfloor$, then it would pass through the (possibly non-integral) extreme point solution of the associated LP. But the use of $\lfloor y_{r0}/y_{rk} \rfloor$ causes the cut constraint to be pushed into the feasible set until it encounters an ILP-feasible vector: in this case, the vector $(x_1, x_2) = (3, 0)$.

The second tableau represents this basic, feasible, all-integer solution. It also shows that x_2 should enter the basis. The execution of a simplex iteration without the addition of a cut would result in a new basic feasible solution $(x_1, x_2) = (3, 1)$ which is integral. However, since the pivot element $y_{12} = 3$ would not have magnitude 1 the method could not have predicted this before pivoting. Therefore, Cut 2 ($3x_1 + x_2 \leq 10$) is appended and a pivot performed. Since Cut 2 does not cut off any of the feasible set, it represents a redundant constraint through the extreme point $(x_1, x_2) = (3, 1)$, and the pivot operation leads to a degenerate solution as shown in the third tableau.

Since the pivot element of the simplex procedure is $y_{1s_1} = 1$ in the third tableau, no appended cut constraint is necessary. But since a basic variable with value zero is exchanged for a non-basic variable with value zero the value of the objective function remains the same. The redundant constraint allows the existence of more than one basic feasible solution at the given extreme point. Since Cuts 3 and 4 also represent redundant constraints through $(x_1, x_2) = (3, 1)$ the solutions found in the remaining

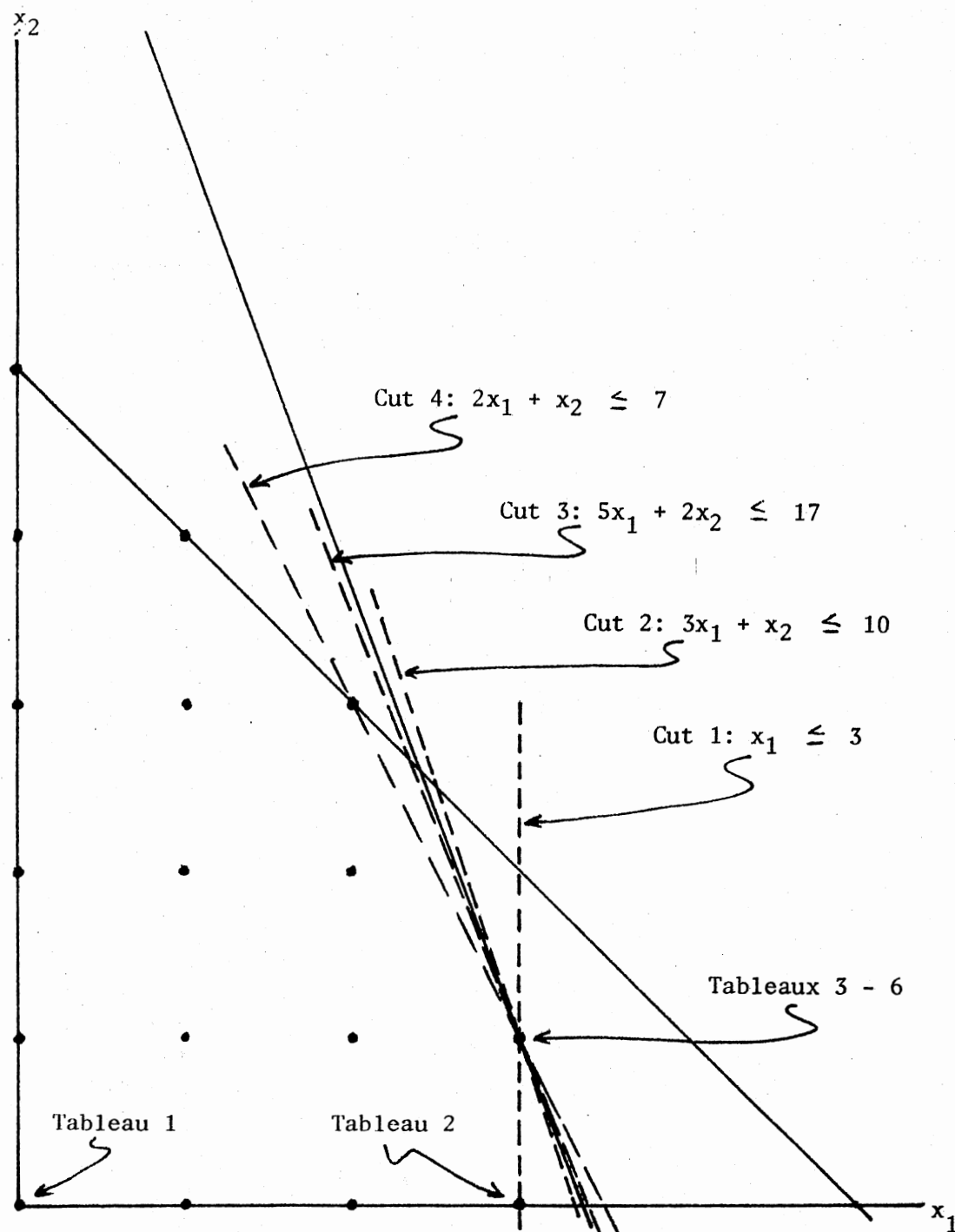


Figure 10. Solutions and Cuts of Example 3.1.1

tableaux are also degenerate.

The four cuts of this example seem to pivot in a counter-clockwise direction about $(x_1, x_2) = (3, 1)$ until the last strikes the ILP-feasible vector $(x_1, x_2) = (2, 3)$. Since the current extreme point solution $(x_1, x_2) = (3, 1)$ lies on the original constraint $8x_1 + 3x_2 + x_3 = 27$, and no other ILP-feasible solution lies on this constraint then the only possibility for an improved integral solution lies off the line $8x_1 + 3x_3 = 27$, that is where x_3 is positive. However, x_3 must be integral when x_1 and x_2 are integral. Therefore, no possibility of moving to an improved ILP-feasible solution exists until a cut is found which would allow x_3 to be both positive and integral. For this reason the optimality conditions are not satisfied until Cut 4 is introduced. In a sense, the method cannot know that its present solution is optimal until it is able to proceed to a new ILP-feasible solution.

Deep Cuts

In a 1969 paper (9) V. Bowman and G. Nemhauser proposed a method for strengthening cuts generated by other methods. The authors referred to these strengthened cuts as "deep cuts". While the idea of strengthening cuts had been used by Wilson with cuts of the All-Integer Method, no general cut strengthening method had been produced prior to that of Bowman and Nemhauser. Their technique will be illustrated first as applied to Gomory's Method of Integer Forms and then in terms of his All-Integer Method.

Recall that the cut of the Method of Integer Forms (2.3.7) can be written:

$$\sum_{j \in R} f_{ij} x_j \geq f_{i0}, \text{ where } f_{ij} = y_{ij} - [y_{ij}].$$

While this cut will not exclude any ILP-feasible vector it may contain one (i.e., there exists some ILP-feasible vector x such that $\sum_{j \in R} f_{ij} x_j = f_{i0}$). However, if the cut (2.3.7) does not contain an ILP-feasible vector, then it may be strengthened in the sense of Definition 2.3.1. That is, there will exist some $t > f_{i0}$ such that

$$\sum_{j \in R} f_{ij} x_j \geq t \quad (3.2.1)$$

is a valid cut.

The strongest valid cut of the form (3.2.1) can be found by locating the largest t such that (3.2.1) is valid; or, alternatively, by finding the smallest t such that $\sum_{j \in R} f_{ij} x_j = t$ is satisfied by some ILP-feasible vector. If the ILP-feasible region is non-empty there will always exist such a t , call it t^* . Thus, $t^* = \text{minimum} \{ \sum_{j \in R} f_{ij} x_j \mid x \in S \}$, where S is the ILP-feasible region. Unfortunately, finding t^* by means of this process is, according to Bowman and Nemhauser "about as difficult as solving the original ILP". However, a trade-off can be made between the strength of the cut and the ease of finding it.

Definition 3.2.1. For any finite $Q \supset S$, where S is the ILP-feasible region, define $t(Q) = \text{minimum} \{ \sum_{j \in R} f_{ij} x_j \mid x \in Q \}$. Since $t(Q)$ is the minimum value over a set containing S , then $t(Q) \leq t^*$, and thus while replacing t^* by $t(Q)$ in (3.2.1) will preserve the cut as valid, the cut will be weaker. The choice of the set Q is made by balancing the ease of finding $t(Q)$ with the strength of the resulting cut. As Q approaches S the corresponding cuts will stronger and $t(Q)$ more difficult to find.

Suppose a canonical maximum ILP

maximize cx

subject to $Ax = b, x \geq 0$ and integer

is being solved by the Method of Integer Forms. The basis for the current iteration is B , and the index set for the current set of non-basic variables is R . Also, the cut (2.3.7) has been derived from the r^{th} row of the tableau (i.e., $i = r$). Let $Q = Q_0 = \{x \mid Ax = b, x_j \geq 0 \text{ and integer for } j \text{ in } R, x_{B_r} \text{ integer}\}$. In other words Q_0 is the ILP-feasible region S with the integrality and non-negativity constraints on the basic variables relaxed, except for integrality on the basic variable x_{B_r} . The r^{th} tableau constraint can be written:

$$x_{B_r} = y_{r0} - \sum_{j \in R} y_{rj} x_j = f_{r0} + [y_{r0}] - \sum_{j \in R} f_{ij} x_j - \sum_{j \in R} [y_{ij}] x_j.$$

And thus,

$$-f_{r0} + \sum_{j \in R} f_{rj} x_j = -x_{B_r} + [y_{r0}] - \sum_{j \in R} [y_{rj}] x_j. \quad (3.2.2)$$

If x is in Q_0 then the right-hand side, and thus the left-hand side of (3.2.2) is integer valued. Therefore, $-f_{r0} + \sum_{j \in R} f_{rj} x_j = p$, an integer. Also, since $f_{rj} \geq 0$, $x_j \geq 0$ for j in R , and $0 \leq f_{r0} < 1$ then $p \geq 0$. So that $t(Q_0) = \text{minimum} \left\{ \sum_{j \in R} f_{ij} x_j \mid x \in Q_0 \right\} = f_{r0} + p^*$, where p^* is the smallest non-negative integer such that $\sum_{j \in R} f_{rj} x_j = p + f_{r0}$ has a non-negative integer solution in the x_j , j in R . Thus, $t(Q_0)$ is found with much greater ease than is t^* .

Example 3.2.1. (from Example 2.3.1) In STEP 3 of Example 2.3.1 the cut $\sum_{j \in R} f_{rj} x_j \geq f_{r0}$ was derived from the x_2 row giving $4/5x_3 + 3/5x_4 \geq 3/5$. The constant in the Bowman-Nemhauser cut replacing $f_{r0} = 3/5$ is

$$t(Q_0) = \text{minimum} \{p + 3/5 \mid 4/5x_3 + 3/5x_4 = p + 3/5 \text{ has}$$

a non-negative integer solution $(x_3, x_4)\}$.

Let $p = 0$, then an appropriate solution exists: $x_3 = 0, x_4 = 1$. Thus, $t(Q_0) = 0 + 3/5 = 3/5$, and the Bowman-Nemhauser cut is identical to the

cut of Example 2.3.1. This situation is implied by the fact, as shown in Figure 7, that the original cut of Example 2.3.1 contains an ILP-feasible vector, and thus cannot be strengthened by increasing the right-hand constant $f_{r0} = 3/5$.

Example 3.2.2. (from Example 2.3.1) Consider the ILP of Example 2.3.1 with the constraint $8x_1 + 3x_2 \leq 27$ modified to $8x_1 = 2x_2 \leq 27$. This modification gives the ILP:

$$\begin{aligned} &\text{maximize} && 700x_1 + 300x_2 \\ &\text{subject to} && 8x_1 + 2x_2 \leq 27 \\ &&& x_1 + x_2 \leq 5, \quad x_1 \text{ and } x_2 \geq 0 \text{ and integer.} \end{aligned}$$

The simplex method applied to the associated LP of this problem gives a final tableau:

		$-x_3$	$-x_4$
x_0	7900/3	400/6	1000/6
x_1	17/6	1/6	-2/6
x_2	13/6	-1/6	8/6

Since the LP optimal solution is not integral, a cut of the form (2.3.7) can be derived from either the x_1 or x_2 rows. If the x_2 row is chosen the cut (2.3.7) becomes:

$$5/6x_3 + 2/6x_4 \geq 1/6. \quad (3.2.3)$$

Now $t(Q_0) = \text{minimum } \{p + 1/6 \mid 5/6x_3 + 2/6x_4 = p + 1/6 \text{ has a non-negative integer solution } (x_3, x_4)\}$. If $p = 0$, it can be seen that no non-negative integral solution exists. But if $p = 1$, a solution $(x_3, x_4) = (1, 1)$ exists. Therefore, $t(Q_0) = 7/6$ and the cut (3.2.3) is strengthened to :

$$5/6x_3 + 2/6x_4 \geq 7/6. \quad (3.2.4)$$

Figure 11 shows the feasible regions along with the cut (3.2.3) and the strengthened cut (3.2.4). It can be seen that (3.2.3) does not contain

an ILP-feasible vector while (3.2.4) does. Also, the "deeper" cut (3.2.4) can be seen to cut off more of the feasible region of the associated LP than the cut (3.2.3). The two cuts can also be seen to be parallel.

If the set Q is further restricted from Q_0 to $Q_1 \subset Q_0$, where $Q_1 \supset S$, then $t(Q_1) \geq t(Q_0)$, and a (possibly) stronger cut results. Let

$$Q_1 = \{x \mid Ax = b, x_j \geq 0 \text{ and integer for } j \text{ in } R \cup \{B_r\}\}.$$

Then $Q_1 \subset Q_0$ since Q_1 is Q_0 with the added constraint that x_{B_r} be non-negative. The set Q_1 can be written as $\{x \mid x \text{ satisfies the constraints (1.5.6), i.e., } x_{B_i} = y_{i0} - \sum_{j \in R} y_{ij}x_j, i = 0, \dots, m; \text{ and } x_{B_r}, x_j \geq 0 \text{ and integer, } j \text{ in } R\}$. But since x_{B_i} may take on any value for $i \neq r$ the only constraint from (1.5.6) which is binding is:

$$x_{B_r} = y_{r0} - \sum_{j \in R} y_{rj}x_j.$$

Thus, $t(Q_1) = \text{minimum} \left\{ \sum_{j \in R} f_{rj}x_j \mid x_{B_r} + \sum_{j \in R} y_{rj}x_j = y_{r0}; x_{B_r}, x_j \geq 0 \text{ and integer, } j \text{ in } R \right\}$. The problem of finding $t(Q_1)$ can now be seen to be equivalent to solving a one-constraint ILP whose variables are x_{B_r} and $x_j, j \text{ in } R$. Thus, if the coefficient matrix A in the definition of Q_1 is $m \times n$, the number of variables in the one-constraint ILP will be $n - m + 1$.

Example 3.2.3. (Bowman and Nemhauser). Suppose the cut corresponding to (2.3.7) is derived from the following row of an ILP tableau:

$$x_3 = 8/10 - 6/10x_1 + 13/10x_2. \quad (3.2.6)$$

The cut derived from this constraint will be:

$$6/10x_1 + 7/10x_2 \geq 8/10. \quad (3.2.7)$$

Now $t(Q_0) = \text{minimum} \{p + 8/10 \mid 6/10x_1 + 7/10x_2 = p + 8/10 \text{ has a non-negative integer solution}\}$. Thus, $t(Q_0) = 18/10$, and the corresponding strengthening of the cut (3.2.7) will be:

$$6/10x_1 + 7/10x_2 \geq 18/10.$$

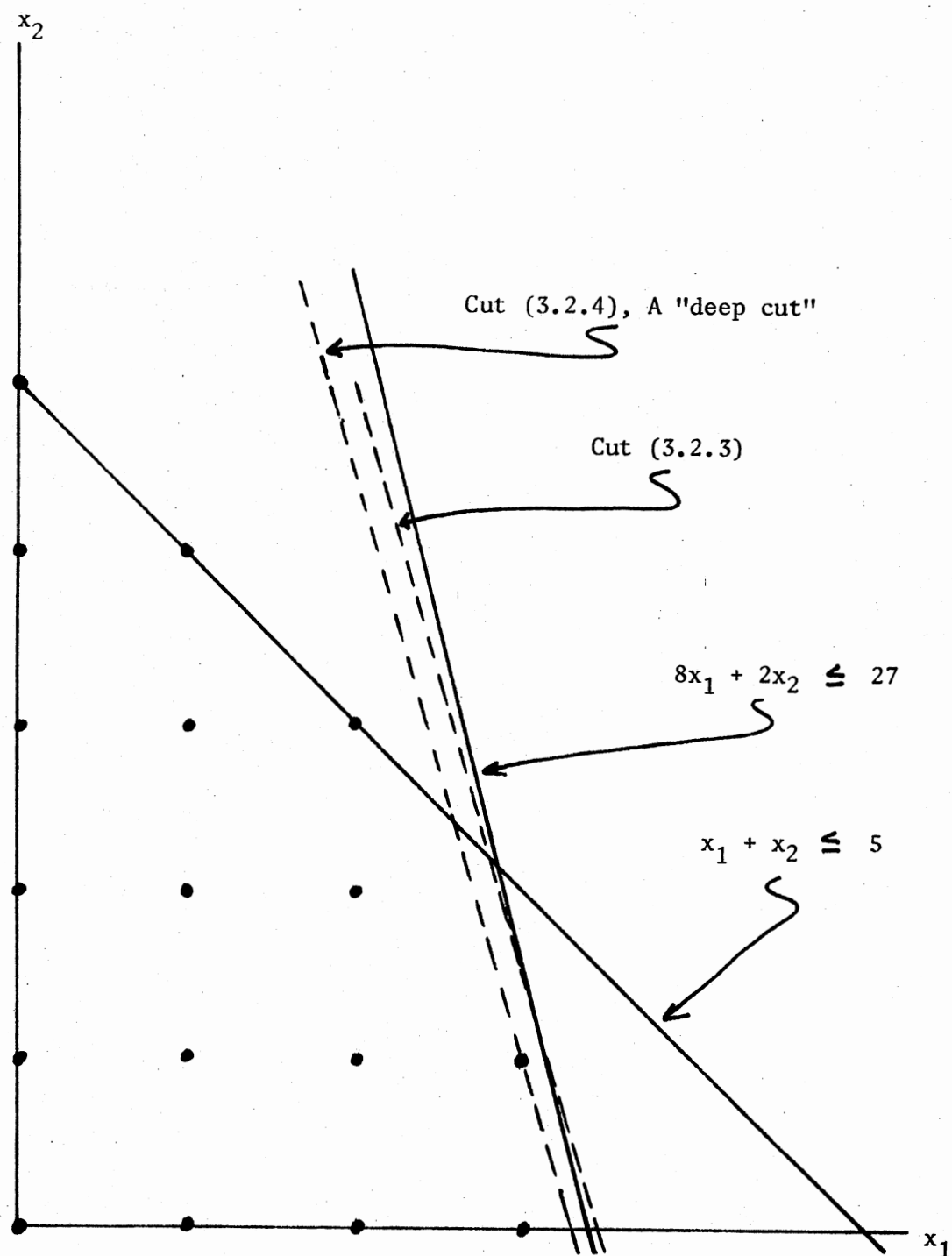


Figure 11. Feasible Region and Cuts of Example 3.2.2

Further, $t(Q_1) = \text{minimum } \{6/10x_1 + 7/10x_2 \mid x_3 + 6/10x_1 - 13/10x_2 + 8/10, x_1, x_2, x_3 \geq 0 \text{ and integer}\}$. Thus, $t(Q_1)$ can be found by inspection to be $t(Q_1) = 28/10$, and the corresponding strengthening of (3.2.7) and thus (3.2.6) will be:

$$6/10x_1 + 7/10x_2 \geq 28/10.$$

Bowman and Nembauser have called cuts derived using $t(Q_0)$ zero-constrained cuts, and cuts derived using $t(Q_1)$ one-constrained cuts. In general, k -constrained cuts are possible where $k \leq m$, the number of constraints in the original ILP. However, the problem of finding $t(Q_k)$ becomes increasingly more difficult as k approaches m .

The one-constraint cut has been applied to Gomory's All-Integer Method with the following modification:

Definition 3.2.2. For any finite $Q \supset S$ and $0 < h < 1$, define $s(Q) = \text{minimum } \{ \sum_{j \in R} - [hy_{rj}]x_j \mid x \text{ in } Q \}$. The object here minimized is the left-hand side of the cut (2.3.18) from the All-Integer Method when h is chosen as in (2.3.19).

With $s(Q)$ replacing $t(Q)$ the cut of the All-Integer Method can (possibly) be strengthened in the same manner as those of the Method of Integer Forms. The following example illustrates the procedure:

Example 3.2.4. (Bowman and Nemhauser). Suppose the following two rows are found in the tableau at some iteration in the All-Integer Method:

$$x_5 = -1 + 2x_1 - x_2 + 5x_3 + x_4 \quad (3.2.8)$$

$$x_6 = -10 - 3x_1 - 5x_2 + 6x_3 + 4x_4 \quad (3.2.9)$$

With $h = 1/2$ the cut (2.3.18) can be derived from the row (3.2.8) as:

$$x_1 + 3x_3 + x_4 \geq 1. \quad (3.2.10)$$

If $x_{B_r} = x_6$ is required to be a non-negative integer then $t(Q_1) =$

minimum $\{x_1 + 3x_3 + x_4 \mid -3x_1 - 5x_2 + 6x_3 + 4x_4 - x_6 = 10, x_1, x_2, x_3, x_4, x_6 \geq 0 \text{ and integer}\}$. A minimum can be found (in this case by inspection, although the ILP could be solved formally) with $x_1 = x_2 = x_3 = 0$, $x_4 = 3$, and $x_6 = 2$. The resulting value of $t(Q_1)$ is 3. Thus, the strengthened cut is:

$$x_1 + 3x_3 + x_4 \geq 3.$$

It may be noted that the strengthened cuts of Bowman and Nemhauser are found by increasing the constant in the cut constraint, while those of Wilson are gotten by changing the value of h in (2.3.18), thus decreasing the values of the coefficients of the x_j while leaving the constant unchanged. It therefore may be possible to combine the two methods to produce even stronger cuts for the All-Integer Method.

The Primal Method of Young and Glover generates a cut containing an ILP-feasible vector. Thus, the cuts of that method cannot be improved by using the techniques developed by Bowman and Nemhauser. Furthermore, any technique similar to that of Wilson for adjusting the value of h in (2.3.18) would be limited by the need for maintaining an all-integer tableau. Thus, the cuts of Young and Glover's Primal Method are the strongest possible given their objective of moving from one ILP-feasible solution to another.

Intersection Cuts

A 1969 paper by Egon Balas (2) described a new type of cut which he called the intersection cut. Unlike previous cuts, intersection cuts were motivated entirely by geometric considerations. In the paper, Balas noted that "the message of this paper lies, not so much in the particular cuts that it proposes, as in the basically new approach to integer programming that these cuts typify." As a result, this work of Balas' was

extended in two 1970 papers: one published jointly by Balas, V. T. Bowman, F. Glover, and D. Sommer (3), and another by F. Glover(23).

The concept of intersection cuts will be introduced by an example solved by the original method of Balas. After that the concept will be discussed in the more general framework of the later papers. Finally, an improvement of the original technique which emerges from the general discussion will be described.

Example 3.3.1. (From Example 1.1.1) The ILP

$$\begin{aligned} &\text{maximize} && 700x_1 + 300x_2 \\ &\text{subject to} && 8x_1 + 3x_2 \leq 27 \\ &&& x_1 + x_2 \leq 5, \quad x_1, x_2 \geq 0 \text{ and integer} \end{aligned}$$

can be solved by Gomory's Method of Integer Forms (Algorithm 2.3.1) with the cut of that method replaced by an intersection cut derived in the following manner:

Pass a circle through the four lattice points defining a unit square containing the current basic feasible solution. This circle will intersect the boundary of the feasible region at two feasible points. The line through these two point defines an intersection cut. The optimal solution $(x_1, x_2) = (12/5, 13/5)$ of Example 1.1.1 is shown in the following tableau and in Figure 12:

		$-x_3$	$-x_4$
x_0	2460	80	60
x_1	12/5	1/5	-3/5
x_2	13/5	-1/5	8/5

The four lattice points defining a square containing $(12/5, 13/5)$ are $(3,3)$, $(2, 3)$, $(2, 2)$, and $(3, 2)$. The circle passing through these four points

has the equation $(x_1 - 5/2)^2 + (x_2 - 5/2)^2 = 1/2$. This circle intersects the boundary of the feasible region at $(2,3)$ and $((357 + 3\sqrt{145})/146, (362 - 8\sqrt{145})/146)$. The second of the two points can be approximated by the rational vector $(27/10, 18/10)$. The intersection cut through the two points is therefore

$$x_2 + 17/10x_1 \leq 64/10.$$

This cut can be represented in terms of the current non-basic variables x_3 and x_4 and slack variable s_1 as

$$s_1 = 14x_3 + 58x_4 - 28. \quad (3.3.1)$$

Appending this constraint to the previous dual simplex tableau gives:

		$-x_3$	$-x_4$
x_0	2460	80	60
x_1	12/5	1/5	-3/5
x_2	13/5	-1/5	8/5
s_1	-28	-14	-58

And after one dual simplex iteration:

		$-x_3$	$-s_1$
x_0	141000/58	3800/58	60/58
x_1	156/58	20/58	-3/290
x_2	106/58	-34/58	8/290
x_4	28/58	14/58	-1/58

The new extreme point solution $(x_1, x_2) = (156/58, 106/58)$ lies inside a square formed by the lattice points $(2,1)$, $(3, 1)$, $(3, 2)$, and $(2, 2)$. The circle passing through these four points has the equation $(x_1 - 5/2)^2 + (x_2 - 3/2)^2 = 1/2$. This circle intersects the boundary of the new feasible set at the point $(3, 1)$ and at a point with irrational coordinates which

can be approximated by $(247/100, 221/100)$. The resulting cut is

$$x_2 + 2.283x_1 \leq 7.849.$$

This cut can be represented in terms of the current non-basic variables as

$$s_2 = 152/((3)(58)) x_3 + 1/58s_1 - 30/58. \quad (3.3.2)$$

The previous tableau with (3.3.2) appended can be written as:

	$-x_3$	$-s_1$	
x_0	141000/58	3800/58	60/58
x_1	156/58	20/58	-3/290
x_2	106/58	-34/58	8/290
x_4	28/58	14/58	-1/58
s_2	-30/58	-152/174	-1/58

After one dual simplex iteration this tableau is replaced by:

		$-x_3$	$-s_2$
x_0	2400	760/58	60
x_1	3	252/290	-3/5
x_2	1	-115/58	8/5
x_4	1	194/174	-1
s_1	30	152/3	-58

This tableau represents an optimal solution of the associated LP, and since it is integral, an optimal solution to the ILP.

Balas (2) has shown that when the coefficient matrix A is m by $n + m$ (2 by 4 in the example) the intersection cut can be given by

$$\sum_{j \in R} (1/\lambda_j) x_j \geq 1, \quad (3.3.3)$$

where the λ_j are computed from the intersections of the hypersphere in R^n

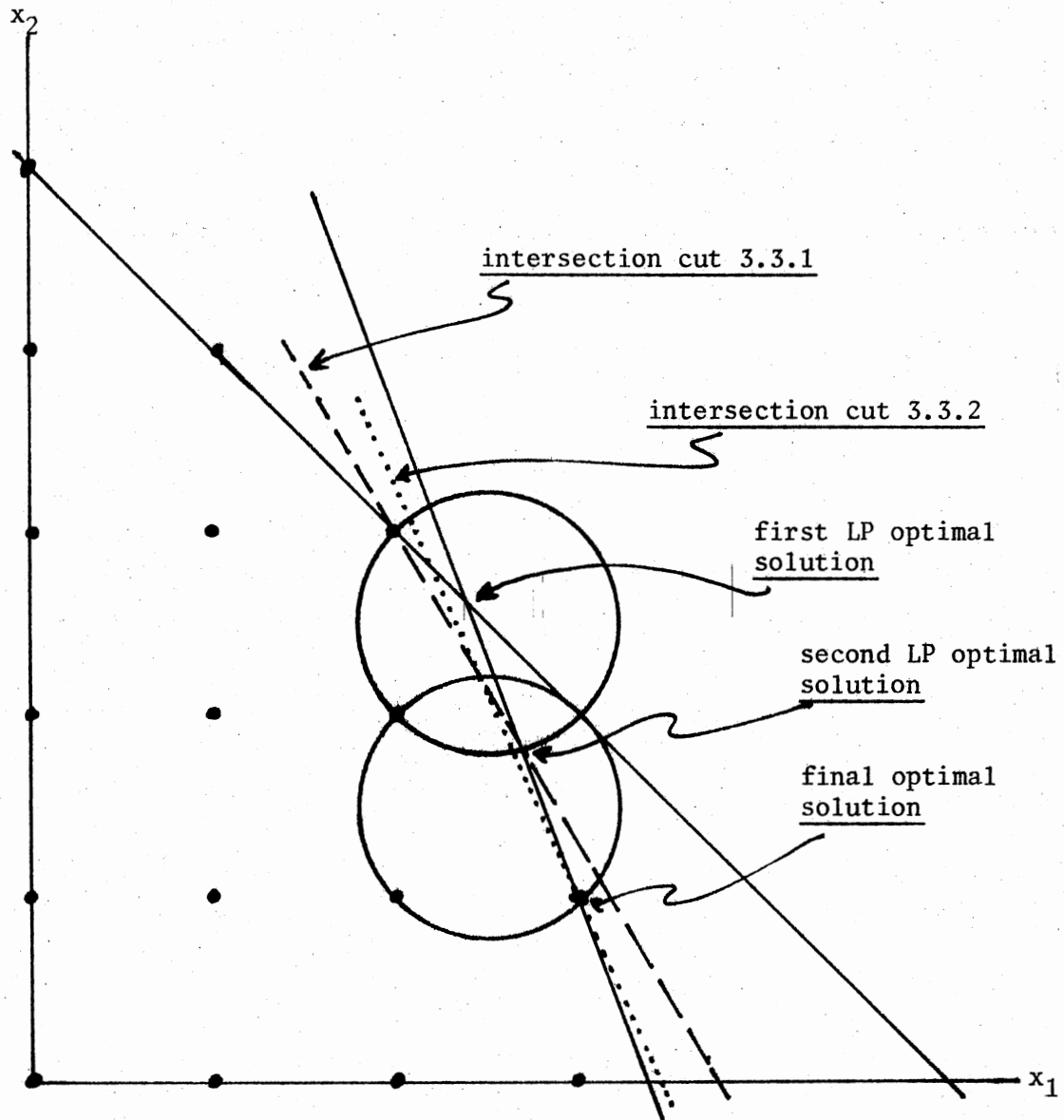


Figure 12. Intersection Cuts for Example 3.3.1

with the feasible region.

The replacement of the Gomory cut with the intersection cut has not improved the method. In fact several problems have been created. One problem is that the algorithm with the intersection cut is not guaranteed to converge. Balas in (2) shows how this problem may be overcome. However, there are two other related problems. First, the intersection cut is difficult to derive from the current extreme point solution; and second, the cut coefficients may be irrational.

These difficulties have led to the modification of Balas' method. In these modifications the hypersphere is replaced by some other convex set with the hope of getting a stronger, more easily computed cut. The following theorem gives conditions under which such a convex set will yield a valid cut:

Theorem 3.3.1. Given an ILP maximization problem with ILP-feasible set $S = \{x \mid \sum_{j \in R} a_{ij}x_j \leq b_i, x_j \geq 0 \text{ and integer, } j = 1, \dots, n\}$ and an associated LP with final simplex tableau represented by

$$x = y_0 - \sum_{j \in R} y_j x_j, \quad (3.3.4)$$

where $x = (x_1, \dots, x_n)$, $y_j = (y_{1j}, \dots, y_{nj})$ for all j , and $y_0 = (y_{10}, \dots, y_{n0})$, and any closed and bounded convex set K in R^n satisfying:

$$y_0 \in K - K_b, \text{ where } K_b \text{ is the boundary of } K, \quad (3.3.6)$$

$$(K - K_b) \cap S = \emptyset, \quad (3.3.7)$$

then for any real $\lambda_j > 0$ such that $y_0 - y_j \lambda_j \in R$ the cut

$$\sum_{j \in R} (1/\lambda_j) x_j \geq 1 \quad (3.3.5)$$

eliminates the current LP optimal solution y_0 while retaining all the vectors in S .

Proof of Theorem 3.3.1. Let K be a closed and bounded set satisfying (3.3.6) and (3.3.7), and let $\lambda_j > 0$, $j \in R$ be given so that $P = \{P_j \mid P_j = y_0 - y_j \lambda_j, j \in R\} \subset K$. Let T be the convex hull of $P \cup \{y_0\}$. That is, $T = \{x \mid x = \alpha_0 y_0 + \sum_{j \in R} \alpha_j P_j, \alpha_0, \alpha_j \geq 0 \text{ and } \alpha_0 + \sum_{j \in R} \alpha_j = 1\}$. Let $T_0 = \{x \mid x \in T \text{ and } \sum_{j \in R} \alpha_j < 1\}$. Let $U = \{x \mid x = y_0 - \sum_{j \in R} y_j x_j, x_j \geq 0\}$, and let $U_0 = \{x \mid x \in U \text{ and } \sum_{j \in R} (1/\lambda_j) x_j < 1\}$. Then $T_0 \subset U_0$ since if $x \in T_0$ then $x = \alpha_0 y_0 + \sum_{j \in R} \alpha_j (y_0 - y_j \lambda_j) = y_0 (\alpha_0 + \sum_{j \in R} \alpha_j) - \sum_{j \in R} \alpha_j \lambda_j y_j = y_0 - \sum_{j \in R} y_j (\alpha_j \lambda_j)$. But $(\alpha_j \lambda_j) \geq 0$, so that $x \in U$. And since $\sum_{j \in R} \alpha_j < 1$, then $\sum_{j \in R} (1/\lambda_j) (\alpha_j \lambda_j) < 1$ so that $x \in U_0$.

Also, $U_0 \subset T_0$ since if $x \in U_0$ then $x = y_0 - \sum_{j \in R} y_j x_j$, where $x_j \geq 0$ and $\sum_{j \in R} (1/\lambda_j) x_j < 1$. Set $\alpha_j = (x_j)/(\lambda_j)$ for $j \in R$, then $\alpha_j \geq 0$ and $\sum_{j \in R} \alpha_j < 1$. Let $\alpha_0 = 1 - \sum_{j \in R} \alpha_j$, then $x = y_0 - \sum_{j \in R} y_j x_j = y_0 - \sum_{j \in R} y_j \alpha_j \lambda_j = y_0 (\alpha_0 + \sum_{j \in R} \alpha_j) - \sum_{j \in R} y_j \alpha_j \lambda_j = \alpha_0 y_0 + \sum_{j \in R} \alpha_j (y_0 - y_j \lambda_j)$ where $\alpha_0 + \sum_{j \in R} \alpha_j = 1$, $\alpha_0, \alpha_j \geq 0$ and $\sum_{j \in R} \alpha_j < 1$. Therefore, $x \in T_0$.

Now $U_0 = T_0 \subset (K - K_b)$. This can be seen by noting that any $x \in T_0$ is a convex combination of y_0 and some point y from the convex hull of P . Since $P \subset K$ then $y \in K$. A standard result from convexity theory [43, p.34] states that if y_0 is in the interior of a convex set K and y is in the closure of K , then any convex combination of y and y_0 will be in the interior of K or will be y itself. Therefore, $x \in (K - K_b)$ or $x = y$. But if $x = y$ then $\alpha_0 = 0$ and $\sum_{j \in R} \alpha_j = 1$, contradicting the fact that $x \in T_0$.

If $x \in S$, then $x \notin (K - K_b)$ since the interior of K contains no lattice point. But if $x \in S$, then by its construction $x \in U$. Therefore, since $U_0 \subset (K - K_b)$, $x \notin U_0$. In other words, if $x \in S$ then $\sum_{j \in R} (1/\lambda_j) x_j \geq 1$, so that the cut (3.3.5) is valid for S . Also, since $y_0 = y_0 - \sum_{j \in R} y_j x_j$, where $x_j = 0$ for every $j \in R$, then $y_0 \in U_0$ and thus satisfies $\sum_{j \in R} (1/\lambda_j) x_j < 1$, so that the cut (3.3.5) cuts off the LP feasible solution y_0 .

This theorem shows that an intersection cut of the form (3.3.5), where the $\lambda_j > 0$ are chosen so the $(y_0 - y_j \lambda_j) \in K$, can be constructed from any closed and bounded convex set K provided K satisfies (3.3.6) and (3.3.7). However, a convex set which contains more of the feasible region will generally produce a stronger cut.

Balas' original cut is constructed in R^n from the convex closed ball $K_2 = \{x \in R^n \mid \|x - ([y_0] + e/2)\|_2 \leq \sqrt{n}/2\}$, where $\|x\|_2 = (\sum_{i=1}^n |x_i|^2)^{1/2}$, $x = (x_1, \dots, x_n)$, and $e = (1, \dots, 1)$. If the ℓ_1 norm instead of the ℓ_2 in R^n is used the following convex set is obtained:

$$K_1 = \{x \in R^n \mid \|x - ([y_0] + e/2)\|_1 \leq n/2\},$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$.

The relationship between the two convex sets K_2 and K_1 can be seen with the aid of Figure 13. Each set contains y_0 in its interior, and the boundary of each set contains the vertices of a hypercube containing y_0 . But since $K_2 \subset K_1$ the intersection of the $(y_0 - y_j x_j)$ and the boundary of K_1 will produce a stronger cut than that produced by K_2 . In addition, since the boundary of K_1 consists of hyperplanes, rather than the surface of a hypersphere, the coefficients of the cut are easier to compute.

This intersection cut produced by the ℓ_1 norm in R^n is described in the paper (3). Other papers describe intersection cuts developed from other norms (10), ellipsoids(11), polytopes (4), and other convex sets (1).

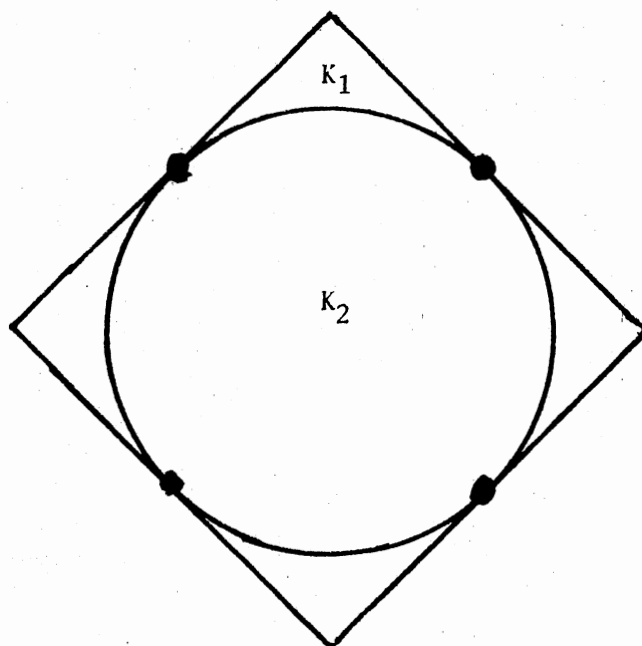


Figure 13. The Relationship between K_1 and K_2

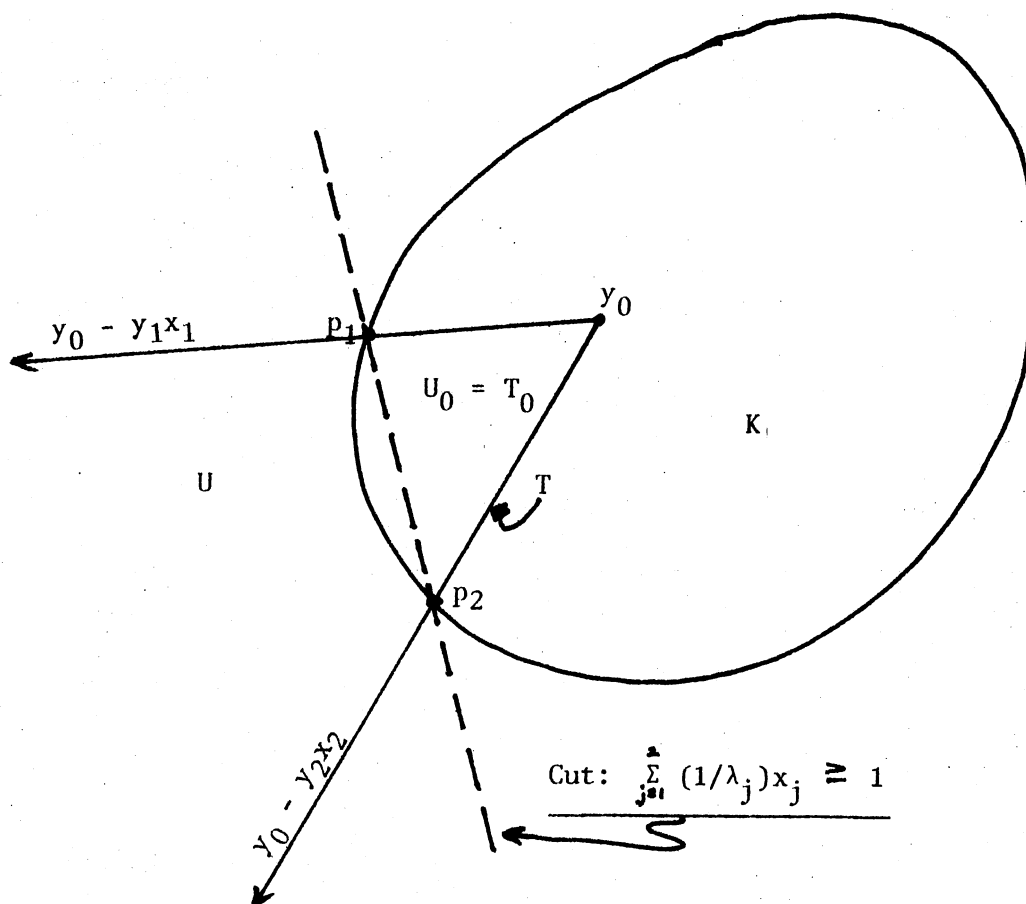


Figure 14. Illustration for the Proof of Theorem 3.3.1

CHAPTER IV

A CHARACTERIZATION OF CUTS

Definitions and Examples

In a 1977 paper R. Jeroslow (33) showed that it could be possible to characterize valid cuts in terms of a certain class of functions called subadditive functions. The purpose of this chapter is to characterize the cuts described in previous chapters using these functions and to show how the application of this idea leads to new cuts.

Definition 4.1.1. A monoid is a set together with a binary operation on that set which is associative and contains an identity element. In the framework of this paper, the set will be some subset of R^n , the binary operation will be vector addition, and the identity element will be zero.

Definition 4.1.2. A subadditive function is a real-valued function defined on a monoid, M , such that $F(a + b) \leq F(a) + F(b)$ for every a and b in M .

The following examples will serve to illustrate the idea of a subadditive function on a monoid, and also to demonstrate that certain functions appearing in later sections are subadditive:

Example 4.1.1. Let the monoid M be R^n and define

$$F(v) = \begin{cases} 0 & \text{if } v = 0, \\ 1 & \text{if } v \neq 0, \end{cases} \quad \text{for every } v \in M. \quad (4.1.1)$$

Now, $F(a + b) > F(a) + F(b)$ only if $(a + b) \neq 0$ while $a = b = 0$. Since

this cannot happen, $F(a + b) \leq F(a) + F(b)$, and thus the function is subadditive.

Example 4.1.2. Let $M = \mathbb{R}^1$ and for any real number h define

$$F_h(v) = [h] v - [vh] \quad (4.1.2)$$

for every $v \in \mathbb{R}^1$. If h, x , and $y \in \mathbb{R}^1$ let $m = [hx]$ and $n = [hy]$. Therefore, there exist p and q such that $0 \leq p < 1$ and $0 \leq q < 1$ and $hx = m + p$ and $hy = n + q$. Since $[m + n + p + q] = m + n$ if $0 \leq p + q < 1$, and $[m + n + p + q] = m + n + 1$ if $1 \leq p + q < 2$, then $[m + n + p + q] \geq m + n$. Therefore, $[hx + hy] = [m + n + p + q] \geq m + n = [hx] + [hy]$, and thus $-[hx + hy] \leq -[hx] - [hy]$. The addition of $[h] \cdot (x + y)$ to both sides of the previous inequality gives $[h](x + y) - [hx + hy] \leq [h]x - [hx] + [h]y - [hy]$, so that $F(x + y) \leq F(x) + F(y)$ for every x and y in M .

Example 4.1.3. Let $M = \mathbb{R}^n$ and let C be a convex set in \mathbb{R}^n containing the origin in its interior. Define

$$F_C(v) = \inf \{ \lambda > 0 \mid v/\lambda \in C \}, \quad (4.1.3)$$

for every $v \in M$. This function is known as the Minkowski functional with respect to C . As noted in (35) this function has several interesting properties. It can be seen from the definition that $F_C(0) = 0$, and that $F_C(v) < 1$ for every v in the interior of C , and $F_C(v) \geq 1$ otherwise. In fact, if C is a closed and bounded convex set, $F_C(v) = 1$ for every v on the boundary of C .

It can be seen that $F_C(\alpha v) = \alpha F_C(v)$ for every real $\alpha > 0$ since $F_C(\alpha v) = \inf \{ \lambda > 0 \mid (\alpha v)/\lambda \in C \} = \inf \{ \lambda > 0 \mid v/(\lambda/\alpha) \in C \} = \inf \{ \alpha(\lambda/\alpha) > 0 \mid v/(\lambda/\alpha) \in C \} = \alpha \inf \{ \lambda/\alpha > 0 \mid v/(\lambda/\alpha) \in C \} = \alpha F_C(v)$.

Finally, $F_C(v)$ is subadditive for every $v \in M$. Let $\varepsilon > 0$ be given and let $x, y \in M$. Choose r and s such that $F_C(x) < r < F_C(x) + \varepsilon$ and $F_C(y) < s < F_C(y) + \varepsilon$. From properties listed above, $F_C(x/r) < 1$ and

$F_C(y/s) < 1$. And so x/r and y/s are in C . Let $q = r + s$. Since C is convex $(r/q)(x/r) + (s/q)(y/s) = (x + y)/q \in C$; so that $F_C((x + y)/q) \leq 1$. Thus, $F_C(x + y) \leq q = r + s < F_C(x) + F_C(y) + 2\varepsilon$. Since ε is an arbitrary positive real then $F_C(x + y) \leq F_C(x) + F_C(y)$.

Example 4.1.4. Let M be the set of lattice points in the non-negative orthant of R^2 under vector addition and define

$$F(v) = \begin{cases} 1 & \text{if } v_1 + v_2 \text{ is an odd integer,} \\ 0 & \text{if } v_1 + v_2 \text{ is an even integer,} \end{cases} \quad (4.1.4)$$

where $v = (v_1, v_2)$. This function is subadditive since $F(v + w) > F(v) + F(w)$ only if $v_1 + v_2 + w_1 + w_2$ is odd while both $v_1 + v_2$ and $w_1 + w_2$ are even, which is impossible.

Jeroslow's Theorems

The following two theorems of R. Jeroslow (33) show that valid cuts can be characterized by a class of subadditive functions in the sense that, given an ILP-feasible set, S , any subadditive function of the class generates a valid cut for S , and any valid cut for S is a weakening of a cut generated by some subadditive function of the class. Statements of these theorems (but not the proofs) can be found in (33).

Theorem 4.2.1. Given a non-empty ILP-feasible set $S = \{x \in R^n \mid Ax = b, x \geq 0 \text{ and integer}\}$ and any subadditive function on the monoid $M = \{v \mid v = Ax \text{ for some } x \geq 0 \text{ and integer}\}$ with $F(0) = 0$, then

$$\sum_{j=1}^n F(a_j)x_j \geq F(b) \quad (4.2.1)$$

holds for every $x \in S$ (i.e., (4.2.1) is a valid cut for S). The a_j are the columns of the matrix A and $x = (x_1, \dots, x_n)$.

Proof of Theorem 4.2.1. Since x_j is a non-negative integer, then for

each $j = 1, \dots, n$; $F(a_j)x_j \geq F(a_jx_j)$ by the subadditivity of F if $x_j > 0$. If $x_j = 0$, then $F(a_j)x_j \geq F(a_jx_j)$ since $F(0) = 0$. Therefore, $\sum_{j=1}^n F(a_j)x_j \geq \sum_{j=1}^n F(a_jx_j)$. Applying the subadditivity of F to the right-hand side of the previous inequality yields $\sum_{j=1}^n F(a_jx_j) \geq F(\sum_{j=1}^n a_jx_j)$. Now, $\sum_{j=1}^n a_jx_j = Ax$, and $Ax = b$ for every $x \in S$, therefore, $\sum_{j=1}^n F(a_j)x_j \geq \sum_{j=1}^n F(a_jx_j) \geq F(\sum_{j=1}^n a_jx_j) = F(Ax) = F(b)$, for every $x \in S$.

Theorem 4.2.2. If $S = \{x \mid Ax = b, x \geq 0 \text{ and integer}\}$ is non-empty and bounded then any valid cut

$$\pi x \geq \pi_0 \quad (4.2.2)$$

for S is a weakening of a cut

$$\sum_{j=1}^n F(a_j)x_j \geq F(b) \quad (4.2.3)$$

for a suitable subadditive function F on the monoid $M = \{v \mid Ax = b \text{ for some integer vector } x \geq 0\}$ with $F(0) = 0$.

Proof of Theorem 4.2.2. Given the cut (4.2.2) valid for $x \in S$ define a function F by

$$F(v) = \inf \{ \pi x \mid Ax = v, x \geq 0 \text{ and integer} \}. \quad (4.2.4)$$

Now $S \neq \emptyset$ and $Ax = b$ for $x \in S$ so that $b \in M$. Since S is bounded and finite it follows that $F(b) = \inf \{ \pi x \mid Ax = b, x \geq 0 \text{ and integer} \}$ is a finite real number.

The next step in the proof is to show that $F(0) = 0$. Let $S_0 = \{y \mid Ay = 0, y \geq 0 \text{ and integer}\}$. Choose b_0 so that $Ab_0 = b$ and $b_0 \geq 0$ and integer. Then $A(b_0 + y) = Ab_0 + Ay = b$ for each $y \in S_0$. Thus, $F(b) \leq \pi(b_0 + y) = \pi b_0 + \pi y$, and so $F(b) - \pi b_0 \leq \pi y$ for each $y \in S_0$. Therefore, $F(b) - \pi b_0 \leq \inf \{ \pi y \mid y \in S_0 \} = F(0)$, and so $F(0)$ is a finite real number. Now since $y \in S_0$ implies $2y \in S_0$, $F(0) = \inf \{ \pi y \mid y \in S_0 \} = \frac{1}{2} \inf \{ \pi(2y) \mid y \in S_0 \} \geq \frac{1}{2} \inf \{ \pi(2y) \mid (2y) \in S_0 \} = \frac{1}{2} F(0)$, so that $F(0) \geq 0$.

But $F(0) \leq 0$ since $0 \in S_0$, so that $F(0) = 0$.

The next step in the proof is to show that $F(v)$ is finite for every $v \in M$. Let $v \in M$; then if $F(v) = r$ where $r \notin \{\pi x \mid Ax = v, x \geq 0 \text{ and integer}\}$, there exists a sequence of infinitely many distinct vectors, $x^{(i)}$, such that $x^{(i)} \geq 0$ and integer, $Ax^{(i)} = v$, and $\pi x^{(i+1)} < \pi x^{(i)}$ for every $x^{(i)}$ in the sequence. Since there are infinitely many distinct vectors in this sequence from the finite-dimensional space R^n there exists some pair $x^{(p)}, x^{(q)}$ with $x^{(p)} - x^{(q)} \geq 0$ and $p > q$. Set $y = x^{(p)} - x^{(q)}$. Since $y \geq 0$ and $Ay = A(x^{(p)} - x^{(q)}) = v - v = 0$ and $\pi y = \pi(x^{(p)} - x^{(q)}) < 0$, then $F(0) = \inf \{\pi x \mid Ax = 0, x \geq 0 \text{ and integer}\} < 0$, which contradicts the conclusion above. Therefore, $F(v) = r \in \{\pi x \mid Ax = v, x \geq 0 \text{ and integer}\}$, so that $F(v)$ is finite for all $v \in M$ and the "inf" in the definition of F can be replaced by "minimum".

Next, it can be shown that (4.2.3) is a strengthening of (4.2.2). Note that $a_j \in M$ for every $j = 1, \dots, n$ since the j^{th} unit vector, u_j , is non-negative and integer and $Au_j = a_j$. For every $j = 1, \dots, n$, $F(a_j) = \min \{\pi x \mid Ax = a_j, x \geq 0 \text{ and integer}\} \leq \pi u_j = \pi_j$. $F(b) = \min \{\pi x \mid Ax = b, x \geq 0 \text{ and integer}\} = \min \{\pi x \mid x \in S\} \geq \pi_0$ since $\pi x \geq \pi_0$ is valid for every $x \in S$.

The final step in the proof involves showing the function F to be subadditive on M . Let v' and v'' belong to M . Choose x' so that $F(v') = \pi x'$, and choose x'' so that $F(v'') = \pi x''$. Thus, $Ax' = v'$ and $Ax'' = v''$, so that $A(x' + x'') = v' + v''$. Therefore, $F(v') + F(v'') = \pi(x' + x'') \geq \min \{x \mid Ax = (v' + v''), x \geq 0 \text{ and integer}\} = F(v' + v'')$.

These two theorems not only demonstrate how the cuts of chapters two and three might be characterized but also provide a method for generating new cuts. For example, consider the function

$$F(v) = \begin{cases} 1 & \text{if } v_1 + v_2 \text{ is odd,} \\ 0 & \text{if } v_1 + v_2 \text{ is even,} \end{cases} \quad (4.2.4)$$

of Example 4.1.4 applied to the ILP of Example 1.1.1 whose ILP-feasible set is described by

$$\begin{pmatrix} 8 & 3 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 27 \\ 5 \end{pmatrix}, \quad x = (x_1, x_2, x_3, x_4) \geq 0$$

and integer. The cut (4.2.1) with F as above would be

$$F((8, 1))x_1 + F((3, 1))x_2 + F((1, 0))x_3 + F((0, 1))x_4 \geq F((27, 5)),$$

or

$$x_1 + x_3 + x_4 \geq 0$$

or in terms of x_1 and x_2 alone

$$4x_1 + x_2 \leq 16. \quad (4.2.5)$$

This cut (as shown in Figure 15) is valid, but not very effective.

If the subadditive function for the cut (4.2.1) is the function F_h of Example 4.1.2 with $h = \frac{1}{4}$, then another cut can be constructed from the feasible region of Example 1.1.1. Let $F(v) = F((v_1, v_2)) = F_h(v_1) = F_{\frac{1}{4}}(v_1) = [\frac{1}{4}]v_1 - [\frac{1}{4}v_1]$, for every $v \in M$. Then the cut (4.2.1) becomes

$$\begin{aligned} & ([\frac{1}{4}] 8 - [8/4])x_1 + ([\frac{1}{4}] 3 - [3/4])x_2 + ([\frac{1}{4}] - [\frac{1}{4}])x_3 \\ & + ([\frac{1}{4}] 0 - [0])x_4 \geq [\frac{1}{4}] 27 - [27/4], \end{aligned}$$

or, in terms of x_1 and x_2 :

$$x_1 \leq 3.$$

As these examples demonstrate, a cut generated by a subadditive function may not be useful even though it does not cut off any ILP-feasible vector. The problem, then, is to determine which function to use given the nature of the ILP-feasible set and the current LP optimal solution. As R. Jeroslow (32) noted, "The key issue of the subadditive

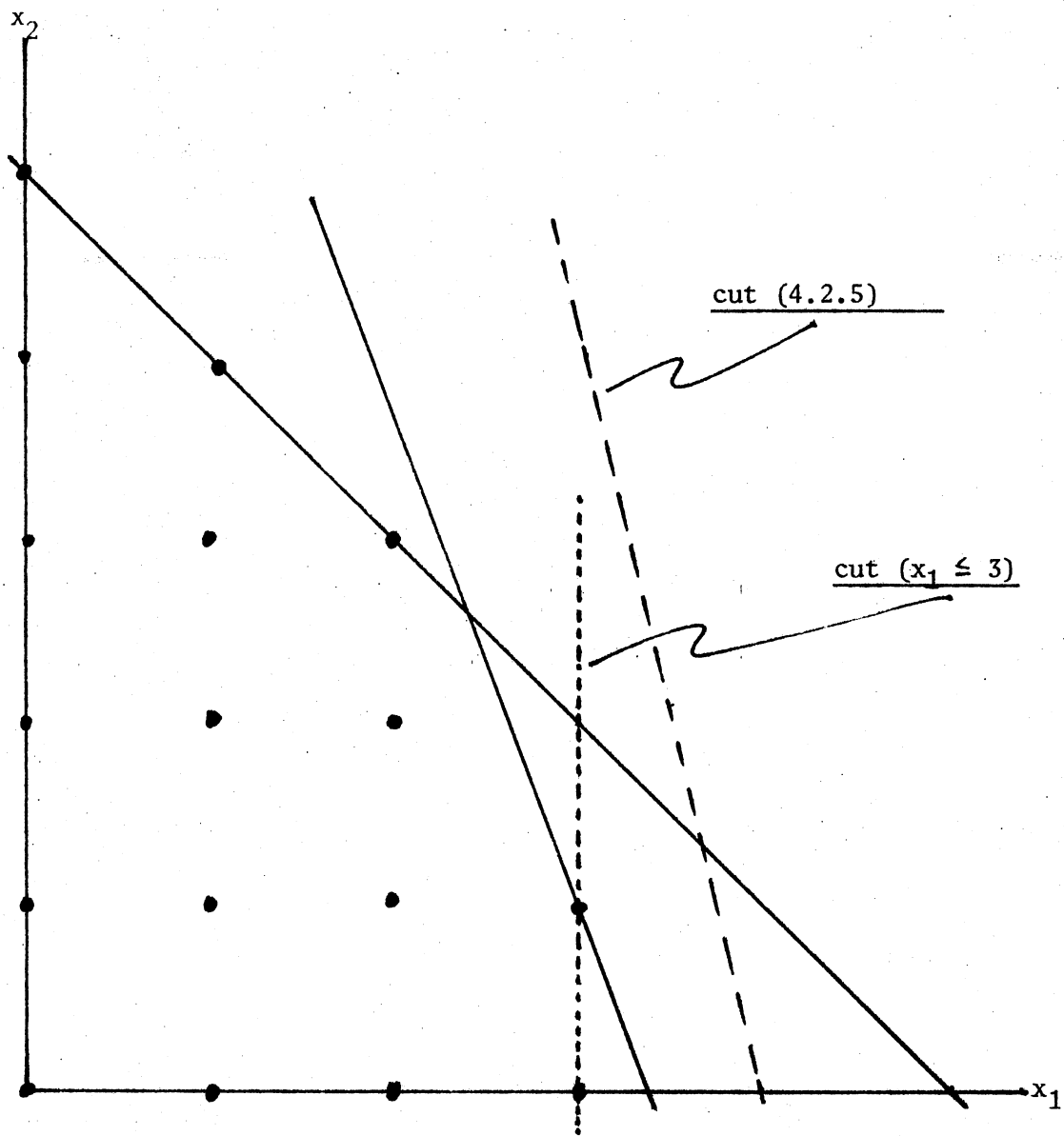


Figure. 15. Subadditive Cuts for Example 1.1.1.

approach is the construction of suitable subadditive functions." In addition to unifying the cuts presented in previous chapters, the characterization of those cuts may lead to further insight concerning which types of subadditive functions are useful.

Characterization of the Cuts

The cuts of chapters two and three can be placed in a unified setting by developing them in terms of their associated subadditive functions. Theorems 4.2.1 and 4.2.2 indicate that subadditive functions can be found which characterize these cuts. The purpose of this section is to present such functions and show how they lead to the corresponding cuts.

The previous cuts were developed from the constraints representing the ILP-feasible region of the given problem (the set $\{x \mid Ax = b, x \geq 0 \text{ and integer}\}$ Theorem 4.2.2). However, the constraint set actually used to get the cut was in the form of the simplex tableau, i.e.,

$$x_B = y_0 - \sum_{j \in R} y_j x_j, \quad (4.3.1)$$

Here x_B represents the vector of basic variables $(x_{B_1}, \dots, x_{B_m})$ and $y_0 = (y_{10}, \dots, y_{n0})$, the current LP-optimal solution. Also, $y_j = (y_{1j}, \dots, y_{nj})$, represents the column vectors of the tableau, and R is the index set of the nonbasic variables. Thus, the subadditive functions used in this section to generate those cuts will be applied to the constraints in the form (4.3.1).

Gomory's Method of Integer Forms (Algorithm 2.3.1) The cut for this method can be derived by using the subadditive function of Example 4.1.2 applied to the constraints (4.3.1) in the form

$$x_B + \sum_{j \in R} y_j x_j = y_0. \quad (4.3.2)$$

In other words, the system of constraints, $Ax = b$, of Theorem 4.2.2 would be

$$(u_1, \dots, u_m; y_1, \dots, y_n) \cdot (x_B; x_N)^T = y_0, \quad (4.3.3)$$

where the u_j are the unit vectors in R^m , the y_j are the n column vectors of the simplex tableau, x_B is the vector of the m basic variables, and x_N is the vector of the n nonbasic variables. Therefore, the a_j of Theorem 4.2.2 are the u_j and y_j of (4.3.3) and the b corresponds to y_0 .

Choose the i^{th} row of the tableau from which to construct the cut.

Let $F(v) = F_h(v_i)$, where $F_h = F_1$ is the function of Example 4.1.2, and v_i is the i^{th} entry of the column v , so that $F(v) = v_i - [v_i]$. The cut (4.2.1) generated by this subadditive function applied to (4.3.3) is

$$\sum_{j \in R} (u_{ij} - [u_{ij}])x_j + \sum_{j \in R} (y_{ij} - [y_{ij}])x_j \geq y_{i0} - [y_{i0}], \quad (4.3.4)$$

but since $u_{ij} = 0$ or 1 , this cut simplifies to

$$\sum_{j \in R} (y_{ij} - [y_{ij}])x_j \geq y_{i0} - [y_{i0}], \quad (4.3.5)$$

which is the cut of Gomory's Method of Integer Forms, equation (2.3.7).

Note that $F(0) = 0$.

Gomory's All-Integer Method (Algorithm 2.3.2) The cut for this method can be derived by using the subadditive function of Example 4.1.2 applied in exactly the same manner as in the derivation above. The only difference being that instead of $h = 1$, $h = \min \{ (-1/y_{rj})[y_{0j}/y_{0k}], j \in R \}$ as in the algorithm. Since Theorem 2.3.2 notes that $0 < h < 1$, the function F_h of Example 4.1.2 becomes

$$F_h(v) = [h]v - [vh] = -[vh]. \quad (4.3.6)$$

Thus, the cut of the form (4.2.1) generated by applying this function to (4.3.3) can be written as

$$\sum_{j \in R} -[hu_{ij}]x_j + \sum_{j \in R} -[hy_{ij}]x_j \geq -[hy_{i0}]. \quad (4.3.7)$$

And since $u_{ij} = 0$ or 1 , the cut can be simplified to

$$\sum_{j \in R} -[hy_{ij}]x_j \geq -[hy_{i0}],$$

which is the cut of the All-Integer Method as in equation (2.3.18). Note again that $F(0) = 0$.

Wilson's Modification of the All-Integer Cut (Example 2.3.5) For Wilson's modification the subadditive function chosen is (4.3.6) above with h^* replacing h , where h^* is chosen as in Example 2.3.5. Since h^* is chosen in such a way that $0 < h^* < 1$, (4.3.6) will correspond to the subadditive function of Example 4.1.2.

The Primal All-Integer Method (Algorithm 3.1.1) The cut for Young and Glover's All-Integer Primal Method can be derived using the subadditive function of Example 4.1.2 applied in the same manner as above. However, as in the Primal algorithm, $h = 1/y_{rk}$. Since $y_{rk} > 1$ then $0 < h < 1$ and

$$F(v) = [h]v - [vh] = -[vh].$$

Thus, this function applied to the constraints (4.3.3) yields

$$\sum_{j \in R} -[hu_{ij}]x_j + \sum_{j \in R} -[hy_{ij}]x_j \geq -[hy_{i0}],$$

or, since $0 < h < 1$ and $h = 1/y_{rk}$,

$$-[y_{ij}/y_{rk}]x_j \geq -[y_{i0}/y_{rk}],$$

which is the cut (3.1.1) of the Primal Method.

Dantzig's Cut (Equation (2.3.17)) To construct Dantzig's cut from a subadditive function first write the constraints of the tableau in the form

$$\sum_{j \in R} y_j x_j = y_0 - x_B. \quad (4.3.8)$$

Therefore, the coefficient matrix, A , of Theorem 4.2.2 becomes (y_1, \dots, y_n) and b becomes $(y_0 - x_B)$. Let the subadditive function be that of Example 4.1.1, i.e.,

$$F(v) = \begin{cases} 0 & \text{if } v = 0, \\ 1 & \text{if } v \neq 0. \end{cases} \quad (4.3.9)$$

The resulting cut is

$$\sum_{j \in R} F(y_j) x_j \geq F(y_0 - x_B). \quad (4.3.10)$$

Now, $(y_0 - x_B) = 0$ only if the current LP optimal solution y_0 , is equal to x_B , and thus integral; in which case no cut is needed. Therefore, $F(y_0 - x_B) = 1$ for all but redundant cuts and $F(y_j) = 0$ or 1 , so that Dantzig's cut, $\sum_{j \in R} x_j \geq 1$, is a weakening of (4.3.10).

Deep Cuts (Chapter 3, section 2) A deep cut of the method of Bowman and Nemhauser is found by taking the cut of the Method of Integer Forms

$$\sum_{j \in R} f_{ij} x_j \geq f_{i0}, \quad (4.3.11)$$

where $f_{ij} = y_{ij} - [y_{ij}]$, and increasing the right-hand side (if possible) to obtain a strengthened cut. The resulting deep cut can be written

$$\sum_{j \in R} f_{ij} x_j \geq t(Q_0), \quad (4.3.12)$$

where $t(Q_0) = \min \{ \sum_{j \in R} f_{ij} x_j \mid Ax = b, x \geq 0 \text{ and integer, } j \in R, x_{B_r} \text{ integer} \}$.

This cut can be characterized by a subadditive function similar to that defined in the proof of Theorem 4.2.2. Let $F(v) = \min \{ \sum_{j \in R} f_{ij} x_j \mid Ax = v, x \geq 0 \text{ and integer} \}$. Then $F(b) = \min \{ \sum_{j \in R} f_{ij} x_j \mid Ax = b, x \geq 0 \text{ and integer} \} \geq \min \{ \sum_{j \in R} f_{ij} x_j \mid Ax = b, x_j \geq 0 \text{ and integer for } j \in R, x_{B_r} \text{ integer} \}$, since the second minimum is taken over a set which contains that of the first minimum. Also, $F(a_j) = \min \{ \sum_{j \in R} f_{ij} x_j \mid Ax = a_j, x \geq 0$

and integer} $\leq \sum_{j \in R} f_{ij} u_j = f_{ij}$ for every $j \in R$.

Thus, $t(Q_0) \leq F(b)$ and $f_{ij} \geq F(a_j)$ for every $j \in R$ so that the cut (4.3.12) is a weakening of the cut (4.2.1) with the function F taken to be that of the proof to Theorem 4.2.2.

Balas' Intersection Cut (Equation (3.3.3)) As was noted in the third section of Chapter 3, Balas' original intersection cut is given by

$$\sum_{j \in R} (1/\lambda_j) x_j \geq 1, \quad (4.3.13)$$

where the λ_j are chosen so that $p_j = y_0 - y_j \lambda_j$ are the (unique) points of intersection of K_b , the boundary of the closed ball K , and the $y_0 - y_j x_j$.

This cut can be described in terms of a subadditive function on the ILP-constraint set (4.3.1) in the form $-\sum_{j \in R} y_j x_j = x_B - y_0$. Thus, the $Ax = b$ of Theorem 4.2.2 is given by

$$(-y_1, \dots, -y_n) x_N = (x_B - y_0). \quad (4.3.14)$$

To construct a suitable subadditive function let K be the closed ball in R^n whose boundary K_b contains the vertices of a hypercube containing y_0 . Let C be the translate of K by y_0 , i.e., $C = \{x \mid x + y_0 \in K\}$. Thus, C is a closed ball containing the origin in its interior (see Figure 16). Let F_C be the Minkowski functional of Example 4.3.1 with respect to the closed convex set C . Then,

$$F_C(v) = \inf \{ \lambda > 0 \mid x/\lambda \in C \}. \quad (4.3.15)$$

F_C is a subadditive function defined on all of R^n with $F(0) = 0$. Since C is a closed ball then for every $v \in R^n$ there exists $\bar{\lambda} > 0$ such that $F_C(v) = \bar{\lambda}$, where $(v/\bar{\lambda}) \in C_b$, the boundary of C . If v is one of the columns of the coefficient matrix in (4.3.14) (i.e., $v = -y_j$ for some $j \in R$) then

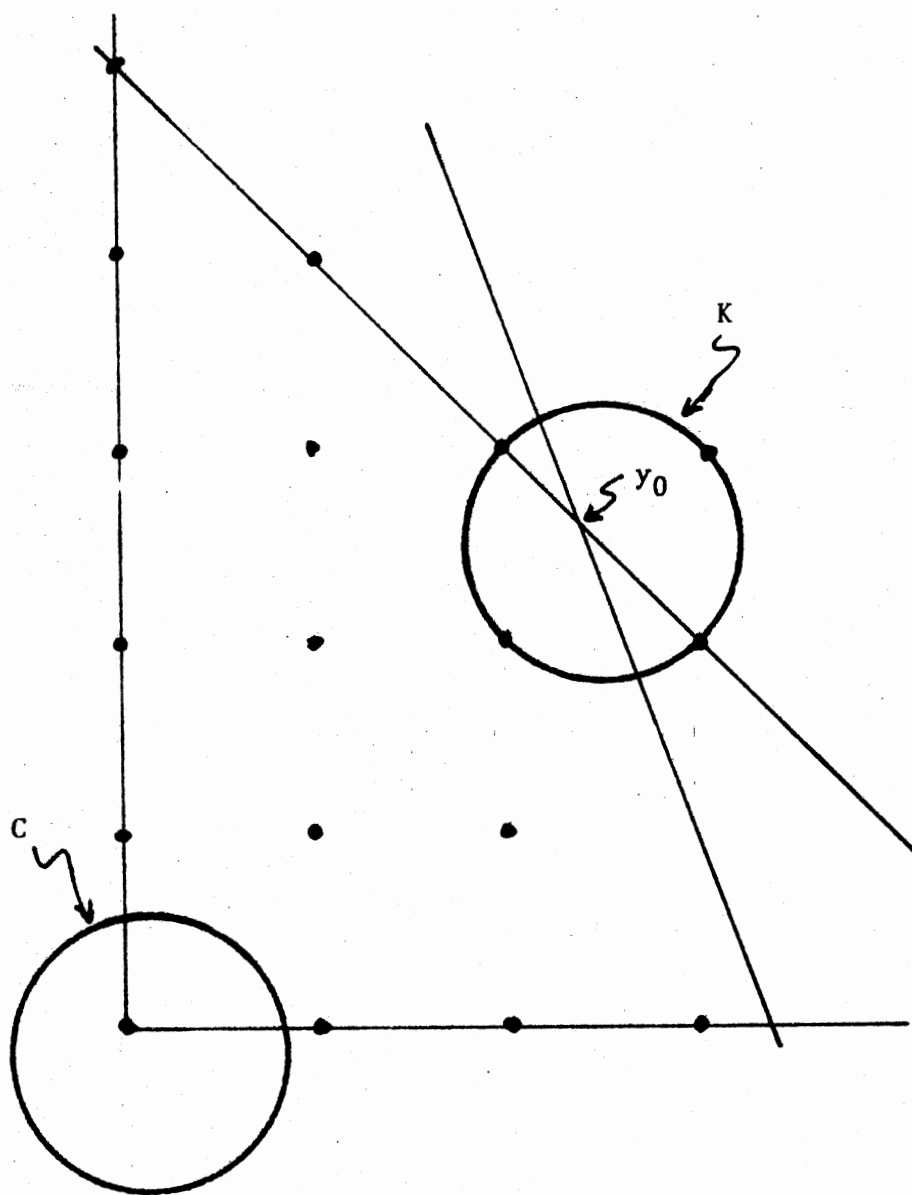


Figure 16. The Convex Balls K and C in \mathbb{R}^2 .

$-y_j/\bar{\lambda}_j \in C_b$. But if $-y_j/\bar{\lambda}_j \in C_b$ then $(y_0 - y_j/\bar{\lambda}_j) \in K_b$. However, the unique point of intersection of $y_0 - y_j x_j$ and K_b is $p_j = y_0 - \lambda_j y_j$.

Therefore, $\lambda_j = 1/\bar{\lambda}_j$, and so $1/\lambda_j = \bar{\lambda}_j = F_C(-y_j)$ for every $j \in R$. Thus, $\sum_{j \in R} (1/\lambda_j) x_j = \sum_{j \in R} F_C(-y_j) x_j$. In other words, the left-hand sides of Balas' intersection cut and the subadditive cut formed using the Minkowski functional with respect to C ,

$$\sum_{j \in R} F_C(-y_j) x_j \geq F_C(x_B - y_0), \quad (4.3.16)$$

are identical. For the subadditive cut (4.3.16) to be a strengthening of Balas' cut (4.3.13) then, it must be true that the right-hand side of (4.3.16) be greater than or equal to that of (4.3.13). It can be seen that $(x_B - y_0)$ does not lie in the interior of C , for if so x_B , an integer vector would lie in the interior of K . But K , by its construction, contains no lattice points. Now since, as noted in Example 4.1.3, for every point $v \notin (C - C_b)$, $F_C(v) \geq 1$, then $F_C(x_B - y_0) \geq 1$ so that the subadditive cut (4.3.16) is a strengthening of the cut (4.3.13).

General Intersection Cuts (Theorem 3.1.1) As shown in Theorem 3.1.1 and closed and bounded convex set K satisfying conditions (3.3.6) and (3.3.7) can be used to produce a valid intersection cut of the form

$$\sum_{j \in R} (1/\lambda_j) x_j \geq 1, \quad (4.3.17)$$

where the λ_j are such that $(y_0 - y_j \lambda_j) \in K_b$, the boundary of the convex set. As in Example 4.1.3, if C is the translate of the convex set K by y_0 , then there exists a Minkowski functional F_C associated with the set C , and therefore by Theorem 4.2.1, a valid cut

$$\sum_{j \in R} F_C(-y_j) x_j \geq F_C(x_B - y_0). \quad (4.3.18)$$

Since C is closed and bounded there exists $\bar{\lambda}_j > 0$ such that $F_C(-y_j) = \bar{\lambda}_j$, where $(1/\bar{\lambda}_j)(-y_j) \in C_b$. Therefore, $(y_0 - (1/\bar{\lambda}_j)y_j) \in K_b$. But since the $(y_0 - y_j x_j)$ intersect K_b at only one point (as can be seen in the proof of Theorem 3.3.1) then $1/\bar{\lambda}_j = \lambda_j$. Thus, as with Balas' original cut the left-hand sides of the subadditive cut (4.3.18) and the intersection cut (4.3.17) are identical.

Also, as with Balas' cut, the left-hand side of (4.3.18) will be greater than or equal to the left-hand side of (4.3.17) since $x_B \notin (K - K_b)$ implies that $(x_B - y_0) \notin (C - C_b)$, and thus $F_C(x_B - y_0) \geq 1$.

Therefore, any intersection cut produced by intersecting a closed and bounded convex set whose interior contains the LP-optimal solution y_0 but no lattice point with the edges of the LP-feasible region $(y_0 - y_j x_j)$ can be characterized as a weakening of a cut produced by applying a (sub-additive) Minkowski function to the columns of the simplex tableau.

A SELECTED BIBLIOGRAPHY

- (1) Balas, E. "Intersection Cuts from Maximal Convex Extensions of of the Ball and the Octahedron." Pittsburg: Carnegie-Mellon University, Management Science Research Report No. 214, 1970.
- (2) Balas, E. "Intersection Cuts - A New Type of Cutting Planes for Integer Programming." Op. Res., Vol. 19 (1971), 19-39.
- (3) Balas, E., V. J. Bowman, F. Glover and D. Sommer. "An Intersection Cut from the Dual of the Unit Hypercube." Op. Res., Vol. 19 (1971).
- (4) Balas, E. and R. Jeroslow. "Canonical Cuts on the Unit Hypercube." Pittsburg: Carnegie-Mellon University, Management Science Research Report No. 198, 1969.
- (5) Balinski, M. L. "On Recent Developments in Integer Programming." Proceedings of the Princeton Symposium on Mathematical Programming. Princeton: Princeton University Press, 1970, 267-302.
- (6) Beale, E. M. L. "A Survey of Integer Programming." Op. Res. Quartley, Vol. 16 (1965), 219-222.
- (7) Ben-Israel, A. and A. Charnes. "On Some Problems in Diophantine Programming." Cahiers Cent. d'Etudes Recherche Operationelle, Vol. 4 (1962), 215-280.
- (8) Blair, C. S. and R. Jeroslow. "The Value Function of a Mixed Integer Program: I." Pittsburg: Carnegie-Mellon University, 1975.
- (9) Bowman, V. J. and G. Nemhauser. "Deep Cuts in Integer Programming." Opsearch, Vol. 8, (1969), 89-111.
- (10) Burdet, C. "On the Algebra and Geometry of Integer Programming Cuts." Pittsburg: Carnegie-Mellon University, Management Science Research Report No. 291, 1972.
- (11) Burdet, C. "Enumerative Cuts: I." Op. Res., Vol. 21 (1973), 63-89.
- (12) Cooper, L. and D. Steinberg. Introduction to Methods of Optimization. Philadelphia: W. B. Saunders, 1970, p. 115.
- (13) Dantzig, G. "Application of the Simplex Method to a Transportation Problem." Activity Analysis of Production and Allocation. Ed. T. C. Koopmans. New York: John Wiley and Sons, 1951, 317-329.

- (14) Dantzig, G. "Discrete Variable Extremum Problems." Op. Res., Vol. 5 (1957), 266-277.
- (15) Dantzig, G. B. "Note on Solving Linear Programs in Integers." Naval Res. Logistics Quarterly, Vol. 6 (1959), 75-76.
- (16) Dantzig, G., R. Fulkerson and S. Johnson. "Solution of a Large-Scale Traveling-Salesman Problem." Op. Res., Vol. 2 (1954), 393-410.
- (17) Feuerman M. and M. Weiss. "A Mathematical Programming Model for Test Construction and Scoring." Man. Sci., Vol. 19 (1973), 961-964.
- (18) Gale, D. The Theory of Linear Economic Models. New York: McGraw-Hill, 1960, p.84.
- (19) Gardner, M. The Unexpected Hanging. New York: Simon and Schuster, 1969, pp. 186-196.
- (20) Geoffrion, A. M. and R. E. Marsten. "Integer Programming Algorithms: A Survey." Man. Sci., Vol. 1 (1972), 465-491.
- (21) Glover, F. "A Pseudo Primal-Dual Integer Programming Algorithm." J. Res. Nat. Bur. Stds., Vol. 17B (1967), 187-195.
- (22) Glover, F. "A New Foundation for a Simplified Primal Integer Programming Algorithm." Op. Res., Vol. 16 (1968), 727-740.
- (23) Glover, F. "Convexity Cuts and Cut Search." Op. Res., Vol. 21 (1973), 123-134.
- (24) Gomory, R. "Outline of an Algorithm for Integer Solutions to Linear Programs." Bull. Amer. Math Soc., Vol. 64 (1958), 275-278.
- (25) Gomory, R. "An Algorithm for the Mixed Integer Problem." RAND Report RM-2597. Santa Monica: The RAND Corporation, 1960.
- (26) Gomory, R. "An Algorithm for Integer Solution to Linear Programs." Recent Advances in Mathematical Programming. Ed. R. Graves and P. Wolfe. New York: McGraw-Hill, 1963, 296-302.
- (27) Gomory, R. "All-Integer Integer Programming Algorithm." Industrial Scheduling. Ed. J. Mutl. Englewood Cliffs, N. J.: Prentice-Hall, 1963, 193-206.
- (28) Gomory, R. and A. J. Hoffman. "On the Convergence of an Integer Programming Process." Naval Res. Logistics Quarterly, Vol. 10 (1963), 121-132.
- (29) Gomory, R. and E. L. Johnson. "Some Continuous Functions Related to Corner Polyhedra." Math. Programming, Vol. 3 (1972), 23-85 and 359-389.
- (30) Heller, I. and C. B. Tompkins. "An Extension of a Theorem of Dantzig's." Linear Inequalities and Related Systems. Princeton: Princeton University Press, 1956, 247-253.

- (31) Hoffman, A. J. and J. B. Kruskal. "Integral Boundary Points of Convex Polyhedra." Linear Inequalities and Related Systems. Princeton: Princeton University Press, 1956, 223-246.
- (32) Jeroslow, R. "Cutting Planes for Relaxations of Integer Programs." Pittsburgh: Carnegie-Mellon University, Management Science Research Report No. 347, 1974.
- (33) Jeroslow, R. "An Introduction to the Theory of Cutting Planes." (Unpublished paper presented to the D077 Conference on Discrete Optimization, Vancouver, B. C., August 1977.) Pittsburgh, Pennsylvania: Carnegie-Mellon University.
- (34) Lemke, C. "The Dual Method of Solving the Linear Programming Problem." Naval Res. Logistics Quarterly. Vol. 1 (1954), 36-47.
- (35) Luenberger, D. Optimization by Vector Space Methods. New York: John Wiley, 1969
- (36) Luenberger, D. Introduction to Linear and Nonlinear Programming. Reading, Mass.: Addison-Wesley, 1973.
- (37) Markowitz, H. M. and A. Manne. "On the Solution of Discrete Programming Problems." Econometric, Vol. 25 (1957), 84-110.
- (38) Martin, G. T. "An Accelerated Euclidean Algorithm for Integer Linear Programming." Recent Advances in Mathematical Programming. Ed. R. Graves and P. Wolfe. New York: McGraw-Hill, 1963, 311-318.
- (39) Neumann, J. von. (Privately Circulated Notes (1947) Reported and Published by D. Gale, H. W. Kuhn and A. W. Tucker in "Linear Programming and the Theory of Games." Activity Analysis of Production and Allocation. Ed. T. C. Koopmans. New York: John Wiley and Sons, 1951, 317-329.)
- (40) Solkin, H. M. "A Brief Survey of Algorithms and Recent Results in Integer Programming." Opsearch, Vol. 10 (1973), 81-123.
- (41) Story, A. E. and H. M. Wagner. "Computational Experience with Integer Programming for Job-Shop Scheduling." Industrial Scheduling. Ed. J. Mutl. Englewood Cliffs, N. J.: Prentice-Hall, 1963, 207-219.
- (42) Trauth, C. A. and R. Woolsey. "Integer Linear Programming: A Study in Computational Effic." Man. Sci., Vol. 15 (1969), 481-493.
- (43) Valentine, F. A., Convex Sets, McGraw-Hill, New York, 1964.
- (44) Wilson, R. B. "Stronger Cuts in Gomory's All-Integer Integer Programming Algorithm." Oper. Res., Vol. 15 (1967), 155-157.
- (45) Young, R. "A Primal (All-Integer) Integer Programming Algorithm." J. Res. Nat. Bur. Stds., Vol. 69B (1965), 213-250.

- (46) Young, R. "A Simplified Primal (All-Integer) Integer Programming Algorithm." Op. Res., Vol. 16 (1968), 750-782.

VITA²

Richard Morton Greenhaw, Jr.

Candidate for the Degree of

Doctor of Education

Thesis: CUT TECHNIQUES IN INTEGER LINEAR PROGRAMMING

Major Field: Higher Education

Minor Field: Mathematics

Biographical:

Personal Data: Born in Sentinel, Oklahoma, July 2, 1948, the son of Richard M. and Betti F. Greenhaw.

Education: Graduated from Edmond High School, Edmond, Oklahoma, May, 1966; received Bachelor of Science degree with a major in Mathematics from Central State University, Edmond, Oklahoma, May, 1970; received Master of Science degree in Mathematics from Oklahoma State University, Stillwater, Oklahoma, May, 1972; completed requirements for Doctor of Education degree at Oklahoma State University, July, 1979.

Professional Experience: NDEA fellow and graduate teaching assistant, Oklahoma State University, Department of Mathematics, Stillwater, Oklahoma, 1970-1973, 1974-1975; actuarial trainee and programmer, Provident Life Insurance Company, Chattanooga, Tennessee, 1973-1974; instructor in mathematics and computer science, Oklahoma Christian College, Oklahoma City, Oklahoma, 1975-1978; assistant professor, Oklahoma Christian College, Department of Mathematics, 1978-1979.

Professional Organizations: Member of Mathematical Association of America.